The MagPi

Issue 94 | June 2020 | magpi.cc | The official Raspberry Pi magazine

8GB
RASPBERRY PI 4
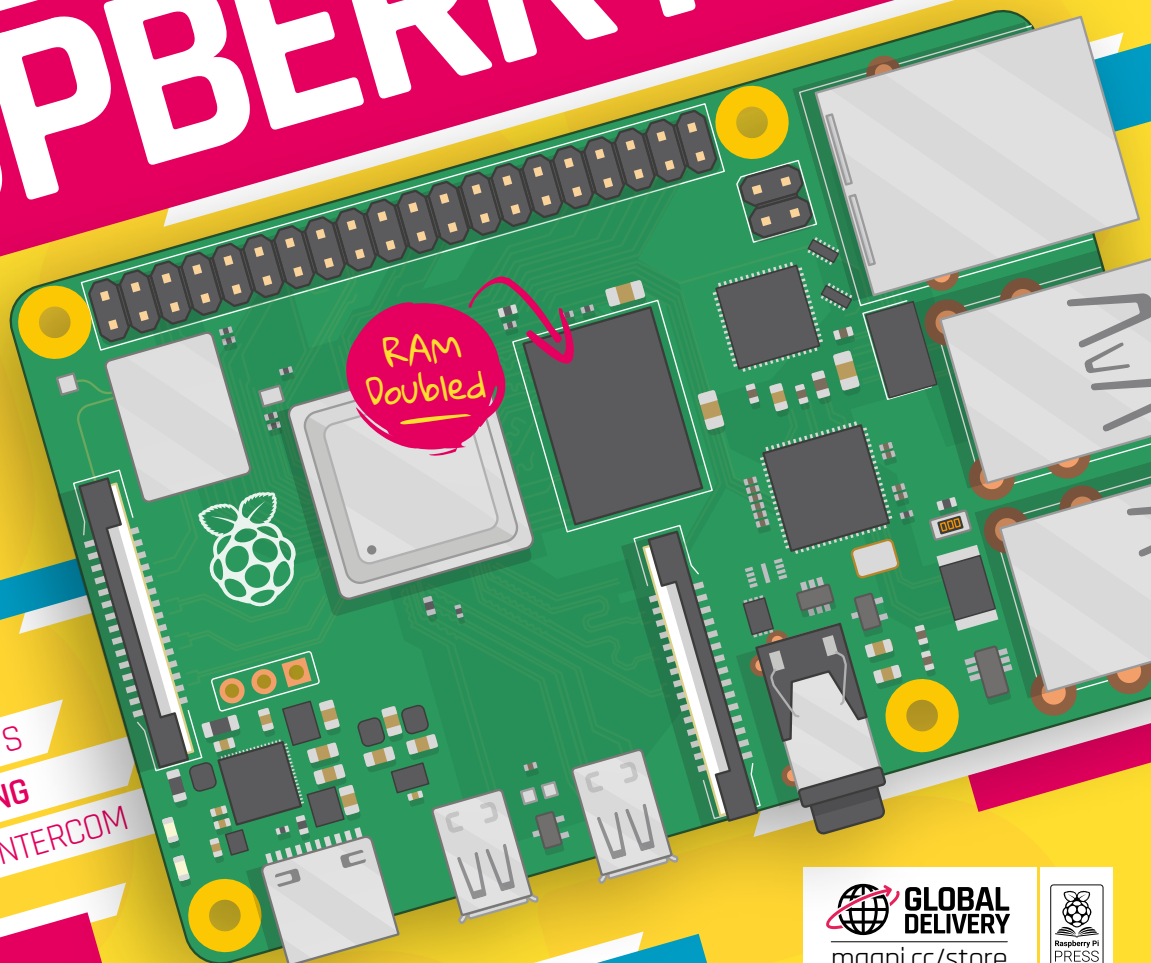
40x THE POWER OF THE ORIGINAL RASPBERRY PI!

RAM Doubled

BUILD A LOW-COST DIGICAM

+ TRACK BEES
10 AMAZING ROBOT KITS
LEARN PYTHON CODING
HOME-MADE VIDEO INTERCOM

GLOBAL DELIVERY
magpi.cc/store

Raspberry Pi PRESS

46 PAGES OF PROJECTS & TUTORIALS

FREE SHIPPING
ON ORDERS OVER £33 OR $50 USD*

Let's make something!

Digi-Key®
ELECTRONICS

0800 587 0991
DIGIKEY.CO.UK

Digi-Key®
ELECTRONICS

9 MILLION+ PRODUCTS ONLINE | **1,000+ INDUSTRY-LEADING SUPPLIERS** | **100% FRANCHISED DISTRIBUTOR**

ECIA MEMBER
Supporting The Authorized Channel

WELCOME
to *The MagPi* 94

For a single-board computer with humble beginnings, the launch of an 8GB model represents an incredible milestone.** It takes Raspberry Pi into new territory, much of which is experimental.

What would you do with a Raspberry Pi with 8GB RAM? That's the question we set Gareth Halfacree, and the result is a fascinating feature (page 36). Gareth looks at options as diverse as full storage cache (where you keep the whole operating system in RAM), through to image and video processing, training machine learning models, and setting up dual-head workstations. There's also an interview with Eben Upton (page 42) which looks at the march into 64-bit processing at Raspberry Pi.

We've had another high-end launch recently, too. Last month we just caught the release of the High Quality Camera. This month we've had a chance to look at it in a little more detail, and we've got some projects for you to look at, along with a guide to making a Raspberry Pi digital SLR-style case (page 66).

If you're new to Raspberry Pi, we have a lot to get you started. From the best robot kits (page 78) to learning Python (page 80), and a huge range of inspirational projects starting on page 12.

Let us know what you plan to make with an 8GB Raspberry Pi and High Quality Camera?

Lucy Hattersley Editor

EDITOR

Lucy Hattersley

Lucy is editor of *The MagPi* and currently in shock at how much it will cost to buy the beloved pinball machine of her youth: Williams' FunHouse.

@LucyHattersley

GET A
RASPBERRY ZERO W KIT
WITH A SUBSCRIPTION!
PAGE 34

Contents

> Issue 94 > June 2020

Cover Feature

Regulars

Project Showcases

36

28

22

Smart Doorbell

BeeMonitor

Tutorials

Make a 3D camera

Rotary encoders – part 3

The Big Feature

High Quality Camera Projects
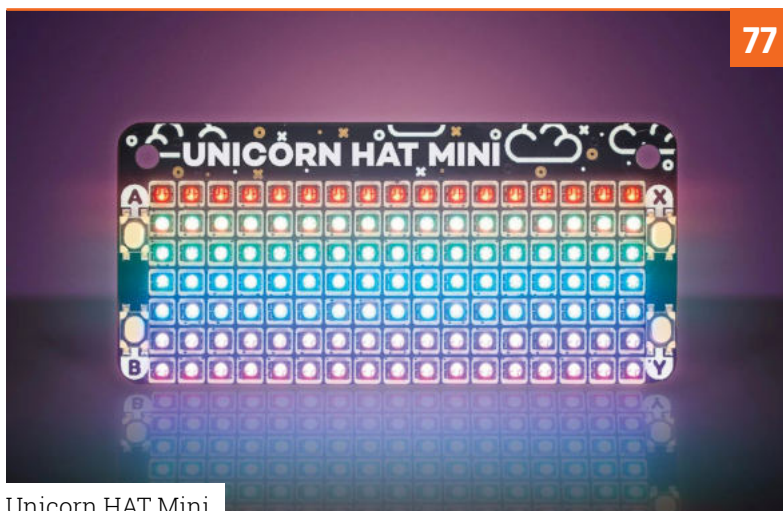
Unicorn HAT Mini

Reviews

Community

Caroline Dunn interview

WIN 1 OF 10 **MAKER PHATS!** 95
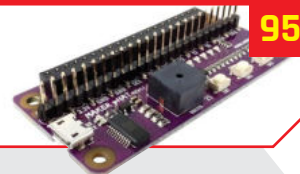
Code
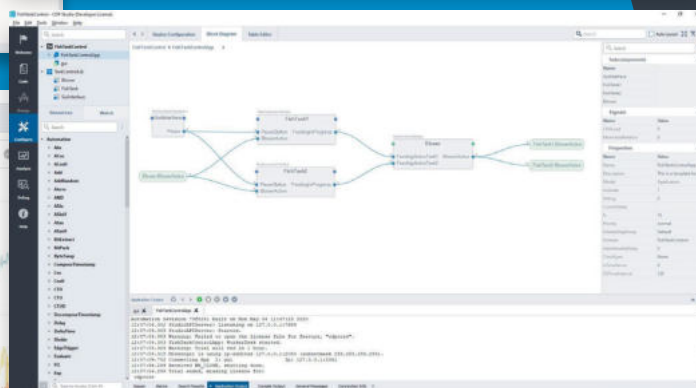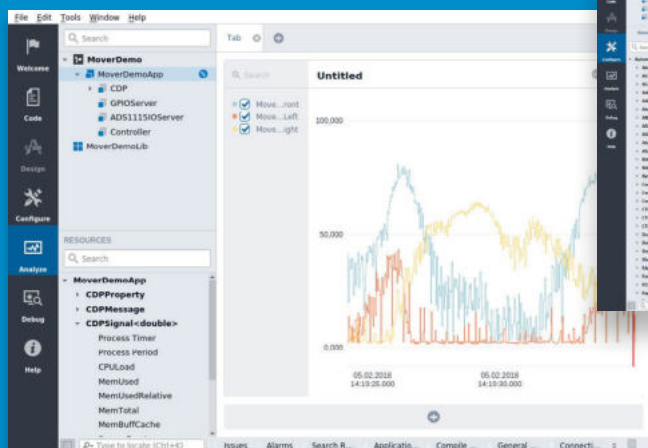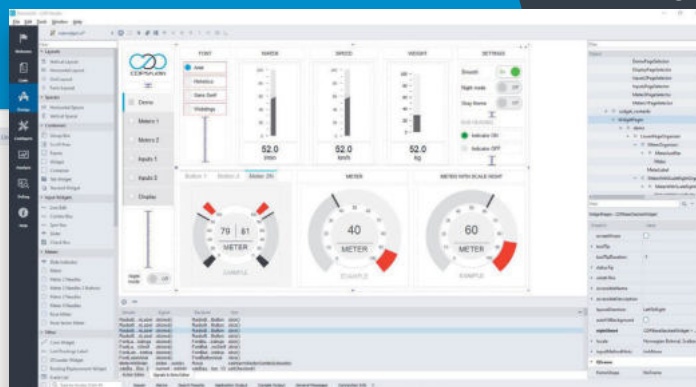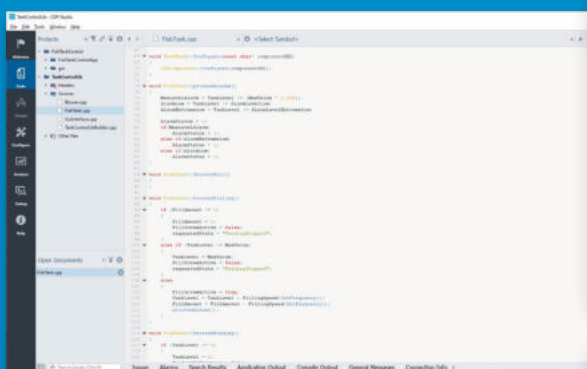
Design

Analyze

Configure
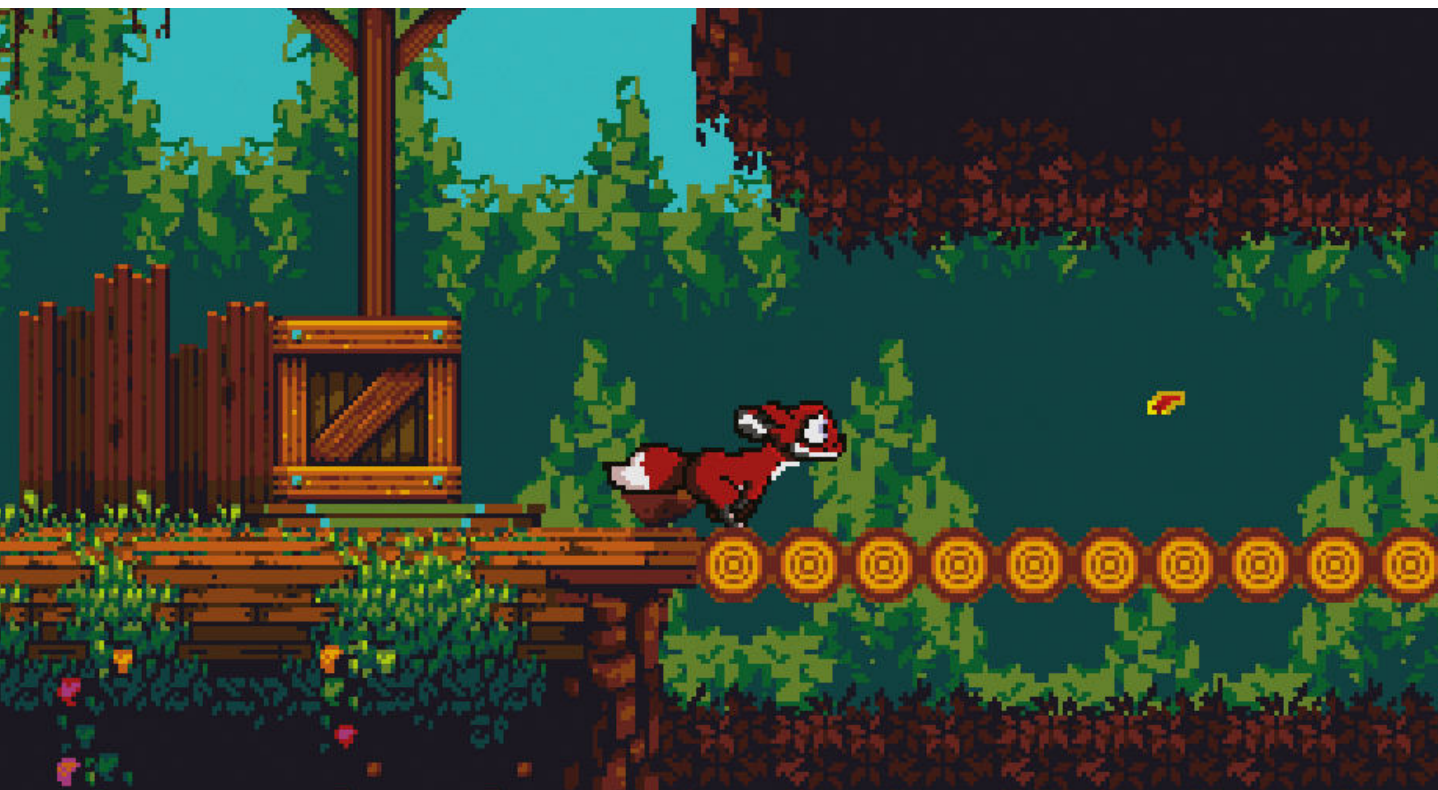
RetroPie gaming gets
a Raspberry Pi 4 boost

Popular games emulator gets dedicated Raspbian Buster build. **Rosie Hattersley** reports

▲ RetroPie offers a
dedicated Raspberry
Pi 4 build of its video
game emulator

Retro video game emulation has just
had a serious boost with the launch
of RetroPie for Raspberry Pi 4 – the
fastest-ever version of the unstoppable
single-board computer.

RetroPie 4.6 sees the games software switch
from the older Raspbian Stretch to Raspbian
Buster – the OS on which Raspberry Pi 4 runs and
which adds support for OpenGL graphics.

Raspberry Pi social media manager Alex Bate
says, "Retro gaming enthusiasts quickly took to
prodding and poking the current version of the
RetroPie software to get it to work on our new,
more powerful computer."

▲ Upcycling and retro gaming go hand in hand,
as GuzziGuy demonstrates

However, less confident RetroPie upgraders have been waiting for a new Raspberry Pi build, hence the excitement about the newly launched version. A spokesperson for RetroPie tells us: "We are marking the Raspberry Pi 4 image as beta for now – primarily as there are still some things to improve on, but most packages run well".

> " Over the years we've seen some marvellous arcade, hand-held, and console builds based around RetroPie "

Different in my day

RetroPie has become synonymous with back-in-the-day gaming and enables people to experience the joy of late 20th century gaming, playing versions of much-loved gaming titles. Rather than simply reminisce about games to today's youngsters, we can build Raspberry Pi and RetroPie-based consoles and hand-held games pads and play the original games.

▲ AdventurePi is a portable RetroPie-based arcade machine

◄ *The MagPi*'s guide to retro gaming includes several RetroPie-based builds
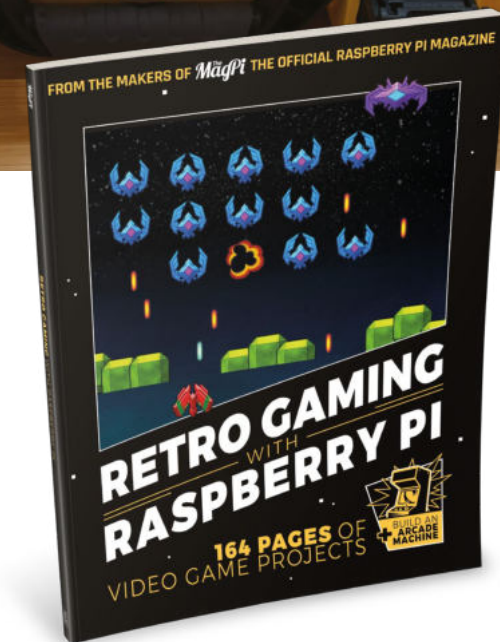
In fact, building a retro gaming rig is one of the most popular first hardware projects for Raspberry Pi enthusiasts. Over the years we've seen some marvellous arcade, hand-held, and console builds based around RetroPie. There's even our dedicated *Retro Gaming with Raspberry Pi* book at **magpi.cc/retrogaming**. To read a step-by-step guide to installing RetroPie for Raspberry Pi, go to **magpi.cc/retropie**. M

Keep on learning

Face-to-face teaching is on pause, but digital literacy is on the up. **Rosie Hattersley** reports

▲ Girls Who Code continue to meet virtually

As we send this issue to print, the UK debate about reopening schools is raging. The lockdown has seen families grapple with helping kids to follow lessons provided by teachers, as well as video-based learning.

While home learning isn't the same as scheduled lessons, there's plenty to suggest alternative modes of learning suit many students. In a weekly seminar hosted by the Raspberry Pi Foundation (**magpi.cc/makingthebest**), Dr Lauren Margulieux, assistant professor of learning sciences at Georgia State University, reported on a number of studies.

The results show that a 'flipped blend' mixture has the best outcome, where content is transmitted via technology, and learning is supported through an instructor.

> **"** Staying focused is a challenge for both parents and children **"**

Feedback loop

Staying focused is a challenge for both parents and children, particularly if you're a little hazy about the content. Asking children to research something or learn a new skill – such as a coding task – independently is worthwhile, but doubly so if you then ask them to tell you about it or even demonstrate what they did.

The same applies to coding: you don't need to have a detailed knowledge of the projects provided by CoderDojo or Girls Who Code. Children's understanding will be enhanced by them undertaking the task and passing on their newfound knowledge to you. There are many courses to help you at **rpf.io/courses**, including how to start your own Code Club. **M**

▶ CoderDojo projects are ideal for family learning

HAL 9000
Google Assistant

A classic sci-fi film was the inspiration for this amazing Raspberry Pi-enabled voice assistant project. **Rosie Hattersley** is impressed

MAKER

Zach Hipps

Electrical engineer Zach is happiest when he's inventing something and says he "loves to inspire others to make things they're excited about"

magpi.cc/bytesized

▼ "I don't think I can do that" – creating HAL's case was far from straightforward and involved rethinking the internal setup

Smart speakers have quickly become assimilated into our everyday lives, with the concept of asking Alexa understood even by those who don't make regular use of tech. Self-confessed tinkerer Zach Hipps wasn't content with a standard-issue voice assistant, however. Instead he chose to give his smart speaker an iconic sci-fi film makeover and the sinister voice of HAL 9000. It's based around the Google Assistant which, he reasoned, already ran on Raspberry Pi and there was likely to be plenty of documentation about it. Having a USB port was another point in Raspberry Pi's favour.

Bits and pieces

One of the most surprising aspects of the HAL 9000 project is how inexpensive it was to build. "I bought a USB microphone, LED, resistor, push-button, and plastic cupcake containers from eBay," reveals Zach. "The remainder of the parts were designed in Fusion 360 and 3D-printed using my Artillery Sidewinder X1 3D printer." Aside from the 3D printing costs, and the Raspberry Pi 3 (which he already had), the project cost about $10.

Zach used the Adafruit Retrogame utility to reprogram a GPIO pin to emulate a keyboard button press. The button, enclosed by the plastic dome from a cupcake, looks impressively like the film version. You press this to ask HAL a question from a list Zach created for the project (see 'Hello HAL' box, overleaf).

"The only piece of code I had to write was a quick Bash script that sources the Python virtual environment and runs the Google Assistant software every time Raspberry Pi boots", he says.

> ❝ Aside from the 3D printing costs, and the Raspberry Pi 3, the project cost about $10 ❞

Perfecting the design

The most time-consuming part of the project was creating and fine-tuning the enclosure for the HAL 9000 Google Assistant so it looked just like the one in the classic film, *2001: A Space Odyssey*. "I imported a photo into Fusion 360 and created a basic sketch. From there I extruded features and added a place for the plastic dome," says Zach.
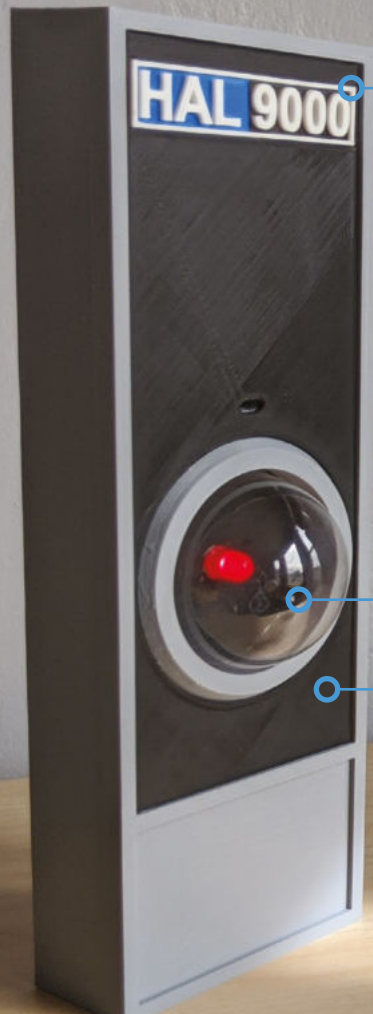
Zach printed the case in multiple parts so he could use several colours of 3D plastic rather than having to paint it afterwards

Connected to a GPIO pin reconfigured using Adafruit Retrograme, the push-button sits beneath a plastic cupcake cover

"I can see you're really upset about this." It's quite strange being greeted by HAL's measured tones

Quick **FACTS**

> Zach grew up building things in the garage with his dad

> He switched to an electrical engineering degree...

> ...soon after discovering Raspberry Pi. Coincidence? Hmm...

> He's an ardent fan of our favourite single-board computer

> He's currently working on a New England covered bridge mailbox notifier
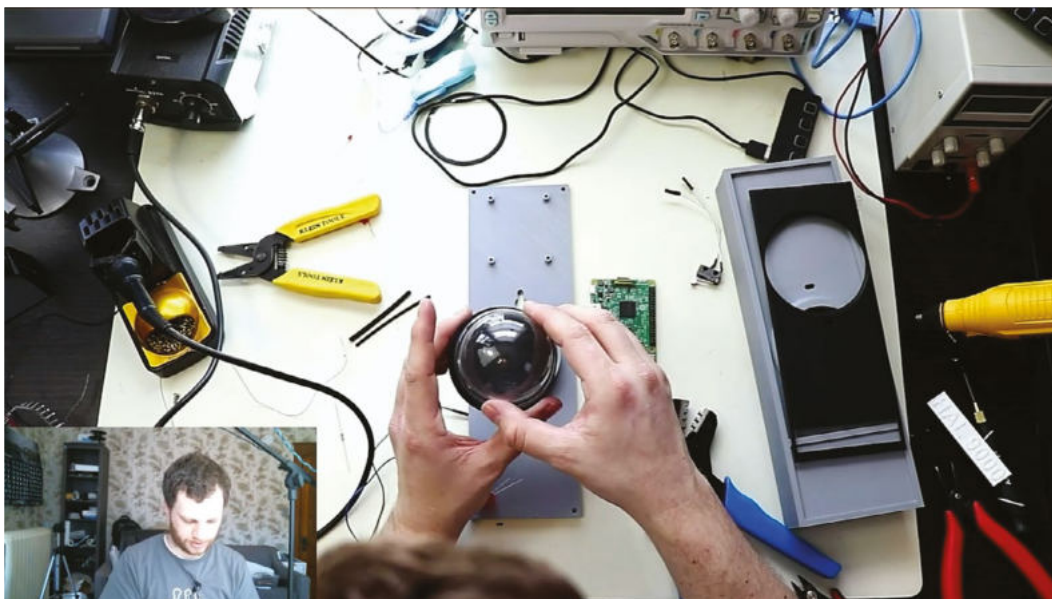
▲ Zach spent hours getting the plastic dome to work as a push-button to activate his HAL 9000 Google Assistant

▲ Designing the case in Fusion 360, ready for 3D printing

Hello HAL

01 To create your own HAL Google Assistant, set the USB microphone and speakers as default playback and capture devices (you'll need device IDs for each). See Zach's hardware setup at **magpi.cc/hal9000ga**.
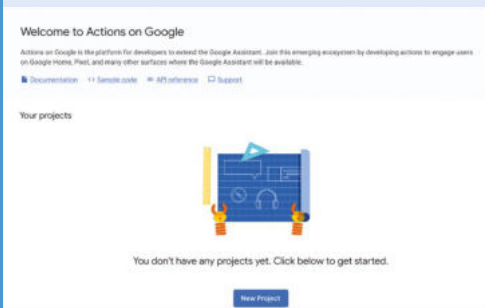
02 Install Google Assistant API on your Raspberry Pi. Create a project in the Google API console (**magpi.cc/googleactions**) and create a Python virtual environment script to run the project. Add a command to make it run on bootup.
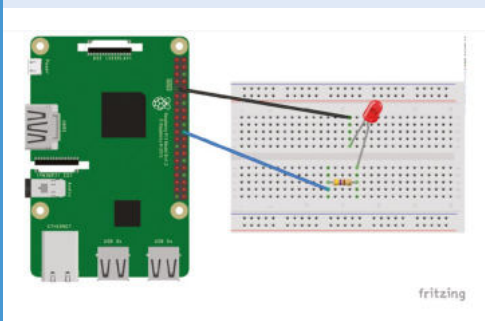
03 Specify a GPIO pin on Raspberry Pi as your 'push-to-talk' button using Adafruit's Retrogame utility (**magpi.cc/retrogame**). Set pressing **ENTER** as the mechanism to activate the talk function. Record a sample question to check it works.

Getting the cupcake dome to work as the push-button was far from plain sailing. After several test prints and iterations, Zach realised the whole thing was simply too large to print as a single piece: "It was about ¾ inch too big to fit on the bed of my 3D printer, so I had to scale it down." However, the enclosure still had to fit several existing components. To make it work, he ended up reversing the position of several items. Raspberry Pi ended up at the top, which then meant the speaker and USB microphone connected to it also needed to be repositioned.

"I had to tweak dimensions until everything fitted correctly," recalls Zach. "A powerful feature of Fusion 360 is that you can design parts parametrically, which means if a test print didn't fit, I could go back and change one or two parameters and the whole design would update."

If a test print didn't fit, I could go back and change one or two parameters and the whole design would update

Once all the pieces had been 3D-printed – a process taking 30 to 40 hours – assembly could begin. Zach positioned the microswitch in its enclosure and installed the 10 mm red LED with its resistor and the clear dome into the mount. He then connected up all the wires to Raspberry Pi. Finally, he put together the all-important casing that makes Zach's HAL 9000 such a showstopper.

"I designed the enclosure to be in several pieces so that I could print them in different colours, rather than having to paint the different colours onto one monolithic print. I glued all these pieces together before putting on the back panel and securing it with screws," he tells us.

"At this point", he says, modestly, "the HAL 9000 Google Assistant using a Raspberry Pi was complete!" **M**

▲ Creating a case that looks just like the original 1968 film version of HAL was a labour of love. Someone suggested it could also make a great photo booth

◀ The finished project. "I am completely operational and all my circuits are functioning perfectly," reports HAL

Game of Life

It may be half-a-century old, but the Game of Life continues to live on thanks to makers such as Nick Kelly, as **David Crookes** explains

MAKER

Nick Kelly

Nick is a software engineer living in San Francisco whose life changed when a friend introduced him to Python and web development.

magpi.cc/ gameoflife

When John Conway unleashed the Game of Life 50 years ago, it caused something of a sensation, particularly among people with an early interest in computing. Described by Conway as a 'non-player' game and modelled to run on a DEC PDP-7 computer, it involved placing a population of cells in a particular pattern on a grid, before running simple rules to create ever-evolving patterns.

John sadly passed away in April this year but his creation lives on, not only in memory but in different manifestations. One of the most intriguing – from our perspective at least – is a version run on a Raspberry Pi. It has been produced by software engineer Nick Kelly, and it uses the same fixed set of rules that has served the simulation well over the years.

"It started as a C# assignment for one of my engineering classes at university, so I had a lot of direction and criteria for what was expected," Nick tells us. "In terms of architecture, I usually take a very object-oriented approach, but since Python was new to me at the time, I went down the functional route."

That's life

The project consists of a Raspberry Pi 2 computer, four 8×8 Adafruit LED grids, and the code. The panels each use Adafruit's FadeCandy, a NeoPixel driver that has built-in dithering and connects to a Raspberry Pi over USB. This allows the LEDs to

▶ You can discover more about John Conway's Game of Life by checking out the large collection of information posted to the LifeWiki (**conwaylife.com**)

▲ English mathematician John Conway invented the Game of Life, which is one of the earliest examples of a cellular automation. He died on 11 April 2020, aged 82
Credit: Thane Plambeck, CC BY 2.0

illuminate or switch off depending on the Game of Life's set of rules.

But what are they? Well, the simple premise is that you have a set of cells, some alive and some dead, behaving in accordance to what is going on around them in the adjoining eight squares. This leads to very complex behaviour and patterns. If there is a live cell with either one or fewer live neighbours, or if there are more than four live neighbours, then that cell will die. If there are two or three neighbours present, then it will remain alive. Dead cells become alive when there are three live neighbours around them, otherwise they remain in a deceased state.

"Designing it wasn't a challenge, but there were plenty of challenges during development," Nick says. "Along with Python, I was also getting a very intense primer to web development and networking. Remote environments, monkey patching, GPIO – these were all foreign terms to me."

Moving on

Delving into a real-world project, he says, provided a perfect way to familiarise himself with a new language. "My boss at work helped me out quite a bit in terms of connecting the technologies

Each LED represents an individual cell which is either dead or alive according to the rules

For this project, a Raspberry Pi 2 was used, connected to the grids via Adafruit's FaceCandy dithering chip

Four 8×8 LED panels make up the 16×16 grid

Quick **FACTS**

> It will run using any Raspberry Pi computer

> The code could work in an infinite space

> The game is governed by four defined laws

> It's about overpopulation and evolution

> Once run, it needs no further interaction

involved, such as Raspberry Pi, LED grids, Open Pixel library, and so on," he says. But Nick also believes improvements can still be made.

"I've refactored code I wrote 24 hours ago, so I would certainly restructure this whole project and add more features," he affirms. "Probably some web interface to select patterns and such."

I would certainly restructure this whole project and add more features

As it stands, however, it's a fun and mesmerising project exploring concepts of underpopulation and reproduction, and Nick has been happy to bring it back to life following John's death. So what has been the end game? "Getting an exotic final pattern has been tough because the project would normally end with four pulsating rectangles," Nick says. But the 'game' runs until the script is stopped and you're never quite sure what you'll see in the meantime. M

▲ To allow the changes to be observed, John Conway originally used a program written by MJT Guy and SR Bourne for a PDP-7 computer with a screen
Credit: Tore Sinding Bekkeda, CC SA 1.0

Apollo Pi

Fans of portable Raspberry Pi projects will love this thermal sensing camera disguised as a 1970s microwave scanner, thinks **Rosie Hattersley**

MAKER

Martin Mander

Martin works in the Intelligence & Analytics department at Norfolk County Council, where he's able to make good use of his love for Internet of Things builds.

kyliemander.com
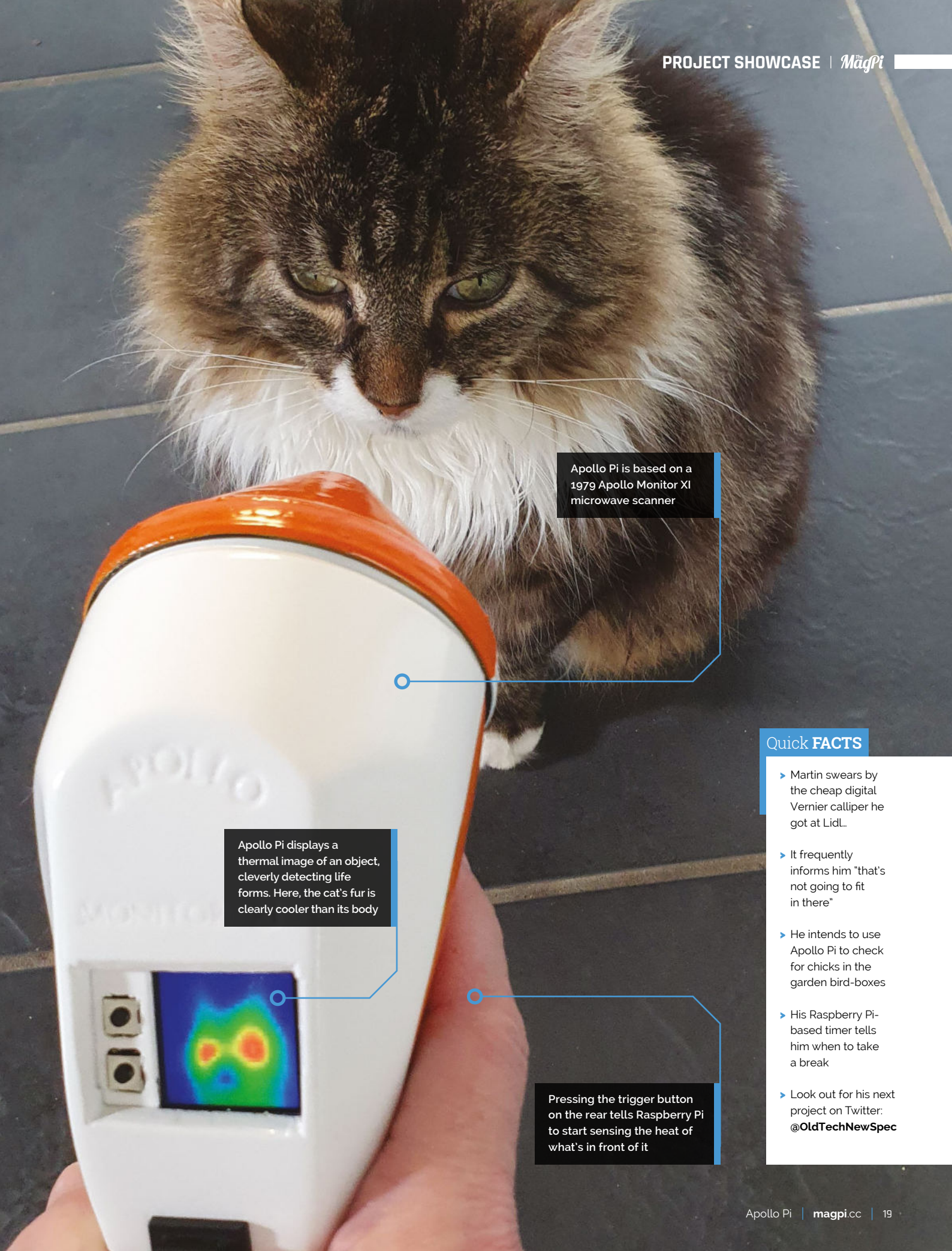
Here at *The MagPi*, we love a retro project and have previously featured several such builds by serial old-tech upcycler Martin Mander. What was to become the latest example – the 1979 Apollo Pi thermal sensing camera – started out as an Arduino project in which Martin and his colleagues at Norfolk County Council challenged each other to design a way of counting library visitor numbers.

When the gauntlet was first laid down, Martin decided it would be a good time to use an Adafruit thermal sensor he'd already bought. "Our libraries already use commercial thermal visitor counters, so I thought I'd build my own," he reasoned. Years of designing projects has taught him that it's best not to get too hung up sticking with your original plans. Instead, he says, it's best "to get started and see where the project takes you!" He advises: "it's always good to do lots of research up front and have a clear idea of what you want to achieve, but some of the most rewarding projects are those where you make at least some of it up as you go along."

▶ The original Apollo Monitor XI microwave scanner Martin adapted for his project

Apollo Pi is based on a 1979 Apollo Monitor XI microwave scanner

Apollo Pi displays a thermal image of an object, cleverly detecting life forms. Here, the cat's fur is clearly cooler than its body

Pressing the trigger button on the rear tells Raspberry Pi to start sensing the heat of what's in front of it

Quick **FACTS**

> Martin swears by the cheap digital Vernier calliper he got at Lidl…

> It frequently informs him "that's not going to fit in there"

> He intends to use Apollo Pi to check for chicks in the garden bird-boxes

> His Raspberry Pi-based timer tells him when to take a break

> Look out for his next project on Twitter: **@OldTechNewSpec**

▶ Testing the thermal sensor and the display settings for the temperature readings

▼ Martin adapted Adafruit's Python code for the thermal sensor. He chose Raspberry Pi because SciPy could use bicubic interpolation to vastly improve the display
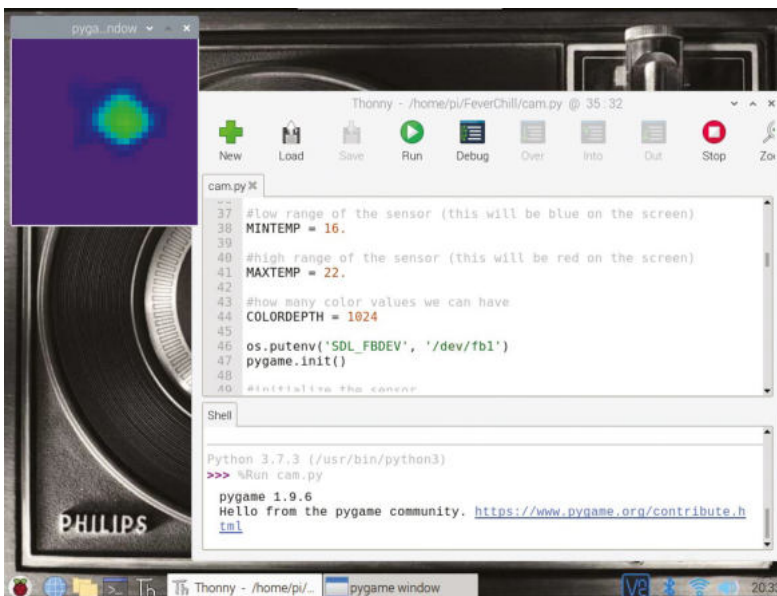
Martin expected it to be a quick build. "I thought I'd pop the components into an old infrared flash gun in an afternoon," he recalls. Unfortunately, the case was too small, so he had to rethink. Digging around in his dusty boxes of old tech for an alternative, the Apollo monitor caught Martin's eye. "It really leapt out at me as it was just the right size, and also a perfect 'hold and point' shape. It was easy to imagine it as a thermal camera."

Change of plan

Parts for the project were sourced online, costing around £70 in all. The biggest expense was the thermal sensor. After connecting the screen, sensor, and Raspberry Pi Zero W, Martin started working out what extra switches he'd need, plus how to power the device. He was able to make use of small bits of leftover plastic as circuit board covers, and spent a long time trimming the original case to make the new components fit.

The screen and sensor are connected to the GPIO pins via jumper cables. To prevent them disconnecting while cramming everything into the grip, Martin hot-glued them in place. With a USB battery pack fitted, he was ready to start testing.

Altering the script

Software for the project (**magpi.cc/apollopi**) is based on Adafruit's Python code for the thermal sensor. "All I've done is added in some extra code to handle more button presses, and some additional integrations with **Adafruit.io**, to enable uploading the captured images and using dashboard sliders to set the temperature threshold," explains Martin, modestly.

> ❝ I thought I'd pop the components into an old infrared flash gun in an afternoon ❞

Apollo Pi took him a month to build, devoting an hour each evening to the project. "I enjoyed taking my time with it and not rushing to finish," he says. As he doesn't have a 3D printer, he had to make everything himself. "A 3D printer would make a much neater job of such tiny parts, but it was incredibly satisfying whittling and filing them by hand!" Ⓜ

▲ Martin chose Apollo Pi's colours to be redolent of the 1970s, and bought the paint just as DIY shops closed down this spring

Build your own
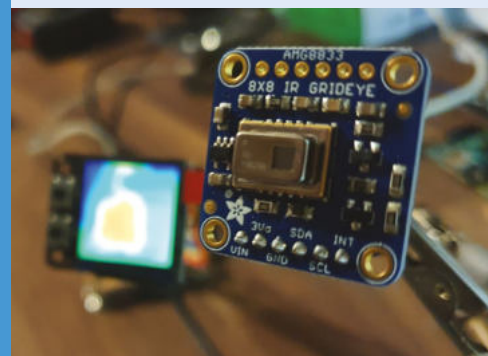
01 Attach the thermal sensor and LCD screen to Raspberry Pi Zero W using jumper cables rather than directly mounting on the GPIO pins.
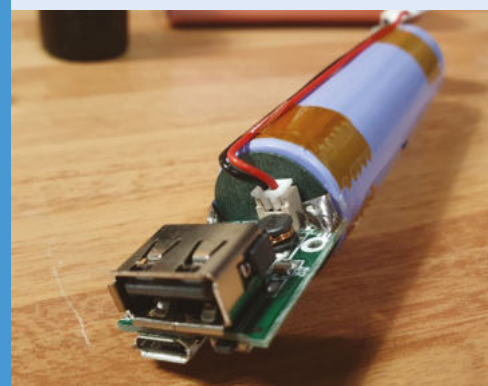
02 Add a slim USB battery pack and connect it to Raspberry Pi Zero W via a cable with a latching button to turn it on/off. Assemble your customised hand-held scanner so any circuitry is hidden, but leave space inside the case so the trigger can be depressed when you want to scan an object.

03 Save instruction code to an FTP site, then use PuTTY and Filezilla to instruct Raspberry Pi to power on the display. Sensor info and code can be found at **magpi.cc/apollopi**.

Smart Doorbell /
Video Intercom System

Have a 'face-to-face' conversation while still guarding your front doorstep with Aaron Tainter's Raspberry Pi-powered twist on the smart doorbell. **David Crookes** calls by

MAKER

Aaron Tainter

Aaron is a software engineer who enjoys making hardware projects in his free time and teaching others how to make them through his YouTube channel, Hacker Shack.

magpi.cc/ hackershack

Smart doorbells have boomed in popularity in recent months, allowing homeowners to see who has dropped by without having to swing open the door. They can, however, be a little one-way: when people ring the vast majority of such devices, you can see them but they can't set eyes on you.

"Not many smart doorbells have a screen for the visitor and I thought creating one would be fun," explains Aaron Tainter. Deciding that his project could also double as a video intercom system if he produced more than one device, he soon got down to work. He figured the system could connect to a

virtual meeting room with the touch of a button and send a notification to a mobile phone, allowing a video connection to be made.

"For this project, I knew that I would need a computer that was small, had input for GPIO pins, could connect to the internet, and could handle video streaming," he reveals. "A Raspberry Pi seemed like the perfect choice – they're really easy to build with and they integrate with everything."

Opportunity rings

Aaron approached the project in his usual way. "I always start by looking at the electronic

> One half of the video chat takes place on this 3.5-inch LCD screen, with the other on the homeowner's mobile phone

> Neatly secreted in the 3D-printed case, the camera captures video of the visitor

> When the button is pressed, a virtual meeting room is opened in Jitsi Meet
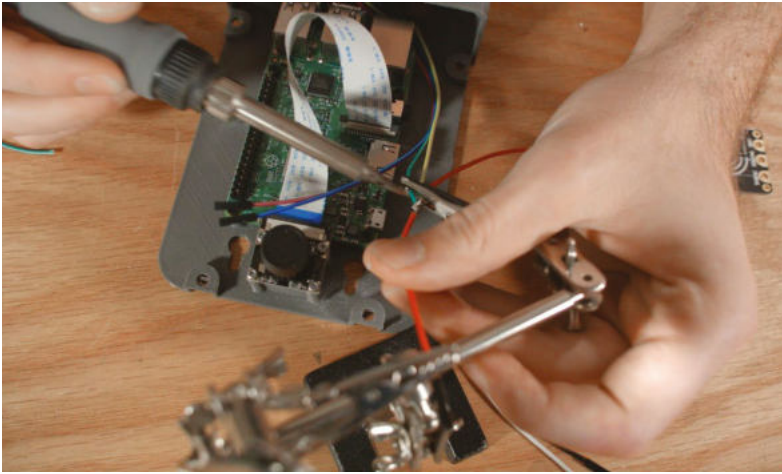
▲ Before attaching Raspberry Pi to the case, Aaron plugged the mic into a USB port. He then connected the camera and speaker, with solder applied to the power wires
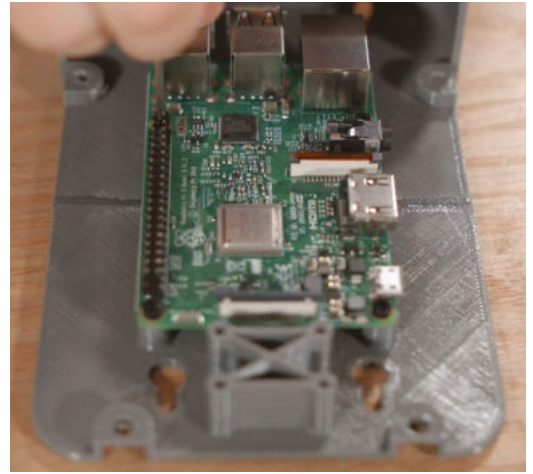
▲ Avoid putting sensitive data on the doorbell's Raspberry Pi because it could be compromised by anyone gaining access to the USB port

components needed for the core features and any open-source software projects that might make the project easier to build," he says.

"Once I've figured out the basic systems design and done some initial experiments with software scripts to verify that the project works, I design a housing to fit around the parts. For this project, the most important feature for me was the audio-video I/O. I had to investigate a few different options for hardware parts to make sure that the audio was good enough for a video call."

The smart doorbell's enclosure was designed using CAD software and it was outputted to a 3D printer. "The most challenging part of this build was finding parts which would all work with the Raspberry Pi for video calls and then getting them to fit inside a small enclosure," Aaron reveals.

"Thankfully, I had my mechanical engineering friend help design the 3D printed enclosure, but it took several revisions to get all the components to fit together after being printed. The tolerances on my printer were great, so we had to do quite a bit of sanding and filing to make everything fit together."

▲ Once everything was fitted, Aaron connected Raspberry Pi over VNC to enable the components and video calling

> ❝ It took several revisions to get all the components to fit together after being printed ❞

Hello world

A momentary button with an LED was fitted along with a Raspberry Pi 3 Model B connected to an LCD screen, camera, USB microphone, and a STEMMA speaker that comes with a plug-and-

play amplifier. Then it was on to the coding. "The software for this project was relatively simple," says Aaron. "It took some experimentation to get all of the electronics configured with the device, but the main script was only a couple of hundred lines." He repurposed some code from an old Raspberry Pi smart security camera that he built a couple of years ago. He used the free encrypted video conferencing app Jitsi Meet (**jitsi.org/jitsi-meet**) to create the video calls.

So how well does it work? "The video stream playback was a little choppy on a Raspberry Pi 3 during extended use, but I think it might have been because the video call website that I used sucked up too much CPU," Aaron reckons. "I haven't tested it, but a Raspberry Pi 4 might work better. People may also want to try implementing their own webcam server and streaming at a lower resolution to optimise performance." ◼

Quick **FACTS**

> ❯ The housing is 3D-printed

> ❯ A chime is emitted when the button is pressed

> ❯ The camera sends footage to a phone

> ❯ Phone footage is sent to the device

> ❯ Aaron says facial detection could be added

Giant Battleships

Play a classic game of Battleships on a giant LED matrix using an antique phone to give orders. **Phil King** dials in

MAKER

Dan Aldred

Raspberry Pi enthusiast, NCCE facilitator, teacher, and coder who enjoys creating new projects and hacks to inspire others to start learning.

tecoed.co.uk

Originally played with a pencil and paper, the Battleships strategy game sees each player positioning a fleet of ships of varying lengths on a grid. They then fire shots in turn, by issuing co-ordinates, to try to hit the opponent's hidden ships and sink them. A board game version featuring plastic pegs for shots was launched in 1967, later followed by snazzy electronic and talking versions.

Dan Aldred has now taken the concept one stage further by recreating the classic game on a giant colour LED matrix, with the player issuing shot co-ordinates using the rotary dial of an old telephone: **magpi.cc/battleships**.

"After creating my giant 10×10 LED board – which is great for light shows, lockdown discos, and general colour displays (**magpi.cc/ledboard**) – I wanted to create a giant game," he reveals. As he had previously coded his own version of Battleships for the Sense HAT, it seemed a natural choice.

In this version, the game is played solo with a limited number of torpedoes to try to sink all the
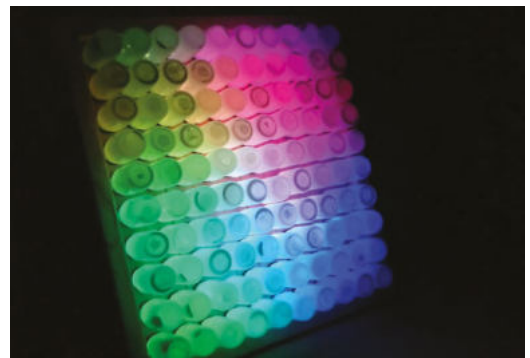
▶ The LED matrix doubles as an impressive disco light show. Dan is considering using it for Conway's Game of Life

randomly placed ships. After a short introduction, the player is asked to select their first co-ordinate. "You simply dial a number and this is then sent as data to the LED board's Raspberry Pi, which triggers another prompt telling the player to dial the second number," explains Dan. "Once the co-ordinates have been dialled in, the board tracks the locations, flashes, and then calculates if the torpedo has hit a boat. If it was successful then a message is sent back to the telephone which triggers a random response consisting of an explosion (sound) and a voice update. If the torpedo misses, different data is sent to the telephone which triggers the miss response, the sound of empty water."

Double helping of Raspberry Pi

While Dan originally intended to use a single Raspberry Pi for the project, he ended up using two of them – one inside the LED matrix and another in the telephone – to avoid the matrix's NeoPixels interfering with the audio for the sound effects and speech, which is played through the telephone's original handset speaker. "The simple solution was to use a second Raspberry Pi which would be situated away from the LED [matrix]."

Two-way communication between the project's two Raspberry Pi boards is achieved using sockets controlled by Python, although Dan encountered

▶ The giant LED matrix comprises a 10×10 grid of NeoPixels, their light diffused by jars

The giant LED matrix comprises a 10×10 grid of NeoPixels, their light diffused by jars

Red cross-hairs show where the shot is targeted – a hit results in an explosion sound effect and flashing lights

Shot co-ordinates are dialled on an old phone; audio effects and instructions are played through its handset speaker
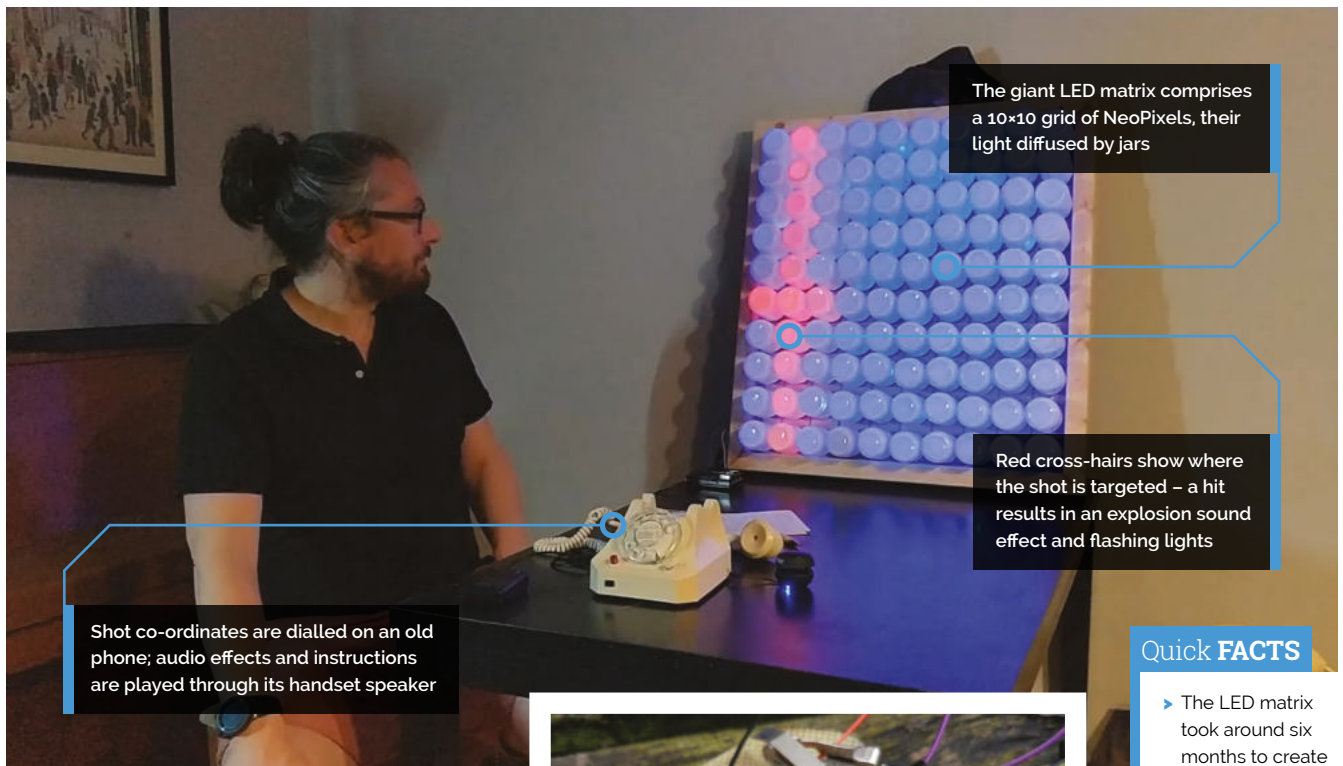
some problems. "I did manage to establish one-way communication between the telephone and the LED board fairly early on in the project but could not get the board to communicate back to the telephone, which meant that I could not trigger the gameplay sounds and updates. Big thanks to Nicole Parrot who showed me that I was using redundant code and sent me an example of the new code and from then on it was plain sailing!"

> ## ❝ You simply dial a number and the data is sent to the LED board's Raspberry Pi ❞

Reading the dial

The method for how the Raspberry Pi reads the numbers from the phone's rotary dial is simple, according to Dan. "Basically, you send a small current from a GPIO pin through the rotary dial and back to a GND pin on Raspberry Pi. When you dial a number, the circuit is broken or interrupted (often referred to as clicks). So if you dial a number one, you get one click, or one interruption; if you dial the number seven, you get seven clicks or interruptions."

Bought for just £5, the 1960s Bakelite phone had plenty of room for housing a Raspberry Pi. Dan discovered it was made by Swiss manufacturer Zellwegger, which used to make government
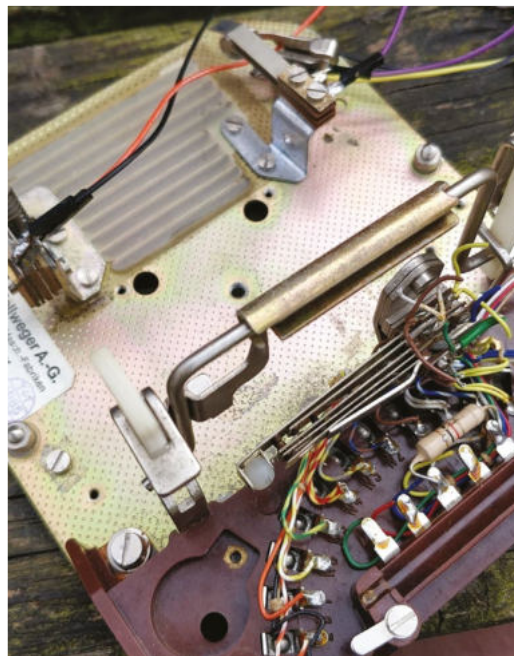
▲ While Dan had used an old GPO phone previously for his Pi Phone project, the mechanism in this Swiss one was completely different

listening devices and phone tapping equipment. "This is where I got excited that perhaps my Swiss telephone had sat on the desk of an important person and been tapped!" says Dan.

While a thorough inspection of all the parts of the phone sadly revealed no bugging device, "There is something quite fun about hearing the instructions and game sounds coming through an old telephone handset." 🔲

Quick **FACTS**

> ➤ The LED matrix took around six months to create

> ➤ Its NeoPixels are diffused by glass jars painted white

> ➤ The Battleships game took a month to develop

> ➤ The phone was bought in an antiques clearance sale

> ➤ Its speaker is wired directly to test pads on Raspberry Pi

Coffee Maker
'Greenhouse'

Upcycling an old coffee machine into a plant-watering gardening assistant sounds like a latte fun. **Nicola King** filters the facts

MAKER

Jeremy Cook

A US-based maker who enjoys tinkering, upcycling, and giving items a new lease of life.

magpi.cc/coffee greenhouseyt

When his wife's ten-year old coffee maker decided to stop working, Jeremy Cook – a maker who likes to upcycle items – saw an opportunity. He felt that the Keurig machine could still have some use, and so he percolated some ideas for its future incarnation. "I couldn't quite bear to part with such a cool piece of technology," he explains. "The Keurig has a water tank and a means of dispensing water, so it seemed only natural to have it water plants."

A beans to an end

Having eventually figured out how to disassemble the original machine, Jeremy then turned his attentions to using its features in a new way. "There's an air pump that, when activated, causes water (or presumably coffee) to go out the spout," he says. "It seems that it pressurises the liquid chamber and forces water out this way... As for my setup, the Raspberry Pi activates this air pump at a certain time. The brew button is a manual

> " The Keurig has a water tank and a means of dispensing water, so it seemed only natural to have it water plants "

trigger for watering. A soil moisture sensor tells the system when it's thirsty (though doesn't control it directly) – when dry, a red light built into the Keurig (and repurposed) lights up. A blue repurposed light signifies that it's watering."

Ingenious, and such a great way to redeploy something destined for the scrap-heap, but Jeremy

Controlled by Raspberry Pi via a transistor, 5 V is applied to the air pump to power the machine

admits that there were a few challenges along the way, including figuring out the button and light circuitry from only the wires: "I didn't disassemble enough to actually see where they were going."

He also admits that it does still require some fine-tuning, but he's generally pretty happy with his new gardening aid. "I left the outer cover off as I think the internals look cool," he adds, "but I hid all the extra parts in the original electronics cavity inside, so it does look mostly original, if partially disassembled."

One person's rubbish...

Jeremy is not done tinkering just yet. "There's an unused but hooked-up power button that I may do some sort of lighting test with," he says. "I suppose it is an automatic watering system, but I've had the thought that I could put a tube on it and water larger

A moisture sensor sends data to Raspberry Pi; if it's too dry, the machine's Brew button is lit up in red

The plant pot was created from a 3D-printed mould

Quick FACTS

> This project took Jeremy 20–30 hours, over a period of around two months

> He wrote an article on the build for arrow.com: **magpi.cc/ coffeegreenhouse**

> A 3D-printed part houses a Raspberry Pi Zero W inside the machine

> He also grows pineapples in pots made from disposable juice containers

> His future project plans include a portable oscilloscope

◀ The exposed rear of the machine, showing the various wires and tubes – it's less complicated than it looks

plants. Again, more of a thought, but you could get like ten of these for a giant watering system. In this case, solar power would be good, but would kind of mess up the looks."

Jeremy chose to use a Raspberry Pi Zero W because it was the perfect size to fit inside the Keurig's electronics compartment, along with a mini breadboard. He has used Raspberry Pi computers before, including running his 3D printer via OctoPrint. "I also have a DIY NAS setup that runs on a Raspberry Pi 4," he adds.

Online feedback on the watering machine has been positive, says Jeremy. "People seemed to think it was a neat idea when I told them about it before it was quite finished, or at least on the internet. I've gotten good feedback from YouTube etc."
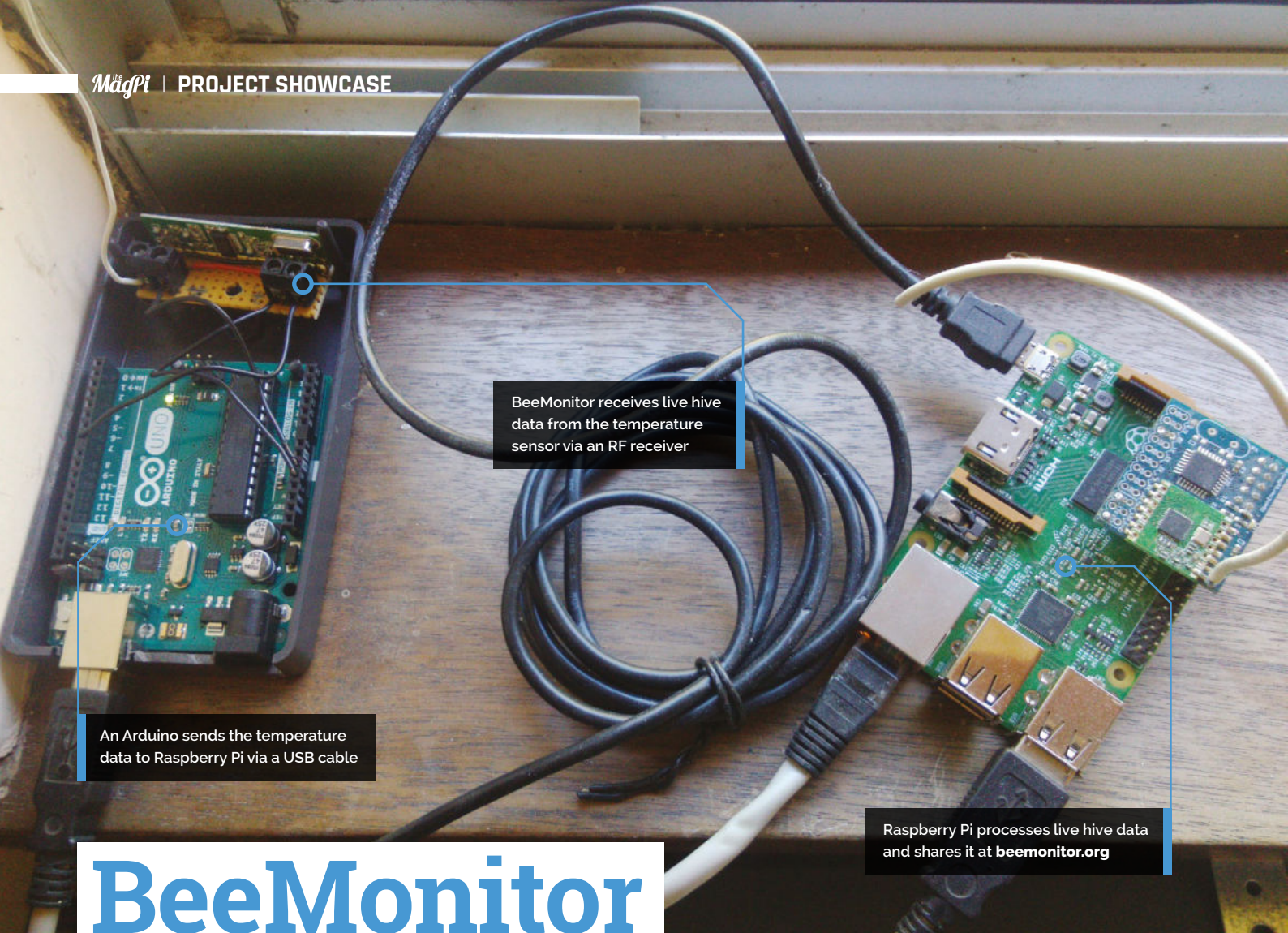
Inspired yet? It just shows what you can do with a little imagination… and a Raspberry Pi, of course! M

BeeMonitor receives live hive data from the temperature sensor via an RF receiver

An Arduino sends the temperature data to Raspberry Pi via a USB cable

Raspberry Pi processes live hive data and shares it at **beemonitor.org**

BeeMonitor

Keeping an eye on bee life cycles is a brilliant example of how Raspberry Pi sensors help us understand the world around us, says **Rosie Hattersley**

MAKER

Glyn Hudson

Having studied electronic engineering at university, Glyn co-founded **openenergymonitor.org**, designing and building open-source energy monitoring tools to help people understand energy and optimise energy consumption and generation.

beemonitor.org

Getting to design and build things for a living sounds like a dream job, especially if it also involves Raspberry Pi and wildlife. Glyn Hudson has always enjoyed making things and set up a company manufacturing open-source energy monitoring tools (**openenergymonitor.org**) shortly after graduating from university. With

▲ Glyn checking the original BeeMonitor setup

access to several hives at his keen apiarist parents' garden in Snowdonia, Glyn set up BeeMonitor using some of the tools he used at work to track the beehives' inhabitants.

"The aim of the project was to put together a system to monitor the health of a bee colony by monitoring the temperature and humidity inside and outside the hive over multiple years," explains Glyn. "Bees need all the help and love they can get at the moment and without them pollinating our plants, we'd struggle to grow crops. They maintain a 34°C core brood temperature (± 0.5°C) even when the ambient temperature drops below freezing. Maintaining this temperature when a brood is present is a key indicator of colony health."

Wi-Fi not spot

BeeMonitor has been tracking the hives' population since 2012 and is one of the earliest examples of a Raspberry Pi project. Glyn built most of the parts for BeeMonitor himself. Open-source

▲ BeeMonitor complete with solar panel to power it. The Snowdonia bees produce 12 to 15 kg of honey per year

software developed for the OpenEnergyMonitor project (**emoncms.org**) provides a data-logging and graphing platform (**magpi.cc/emoncms**) that can be viewed online.

The hives were too far from the house for WiFi to reach, so Glyn used a low-power RF sensor connected to an Arduino which was placed inside the hive to take readings. These were received by a Raspberry Pi connected to the internet.

At first, there was both a DS18B20 temperature sensor and a DHT22 humidity sensor inside the beehive, along with the Arduino (setup info at **magpi.cc/ds18b20sensing**). Data from these was saved to an SD card, the obvious drawback being that this didn't display real-time data readings. In his initial setup, Glyn also had to extract and analyse the CSV data himself. "This was very time-consuming but did result in some interesting data [**magpi.cc/beemonitorv1**]," he says.

" Bees need all the help and love they can get at the moment "

Sensor-y overload

Almost as soon as BeeMonitor was running successfully, Glyn realised he wanted to make the data live on the internet. This would enable him to view live beehive data from anywhere and also allow other people to engage in the data.

"This is when Raspberry Pi came into its own," he says. He also decided to drop the DHT22

humidity sensor. "It used a lot of power and the bees didn't like it – they kept covering the sensor in wax! Oddly, the bees don't seem to mind the DS218B20 temperature sensor, presumably since it's a round metal object compared to the plastic grille of the DHT22," notes Glyn.

The system has been running for eight years with minimal intervention and is powered by an old car battery and a small solar PV panel. Running costs are negligible: "Raspberry Pi is perfect for getting projects like this up and running quickly and reliably using very little power," says Glyn. He chose it because of the community behind the hardware. "That was one of Raspberry Pi's greatest assets and what attracted me to the platform, as well as the competitive price point!" The whole setup cost him about £50.

Glyn tells us we could set up a basic monitor using Raspberry Pi, a DS28B20 temperature sensor, a battery pack, and a solar panel. 𝗠

Monitoring set-up on National crown board
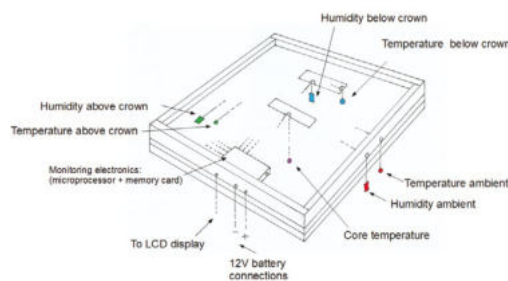
▲ Unlike the humidity sensor, the bees don't seem to mind the temperature probe

◄ Diagram showing what information BeeMonitor is trying to establish

Raspberry Pi
loft bed

Upgrading a bed into a childhood dream
is apparently easy with Raspberry Pi, as
a sleepy **Rob Zwetsloot** found out

MAKER

Jolien Oomens

A mathematics graduate currently working as a data scientist who loves to do DIY projects merging the digital and physical worlds.

Being inside a lot can easily get you down, especially if you don't seem to have gained any free time. For people that do have a little extra time, a world of possibilities is open if you want to take it. Like Jolien Oomens, who built an entire bed and added a Raspberry Pi to it.

"This March, two friends of mine organised a Raspberry Jam in my hometown, Amsterdam." Jolien explains. "There I saw some really cool projects. Living in Amsterdam, more living space in my apartment is always very welcome and I've wanted a loft bed for some time."

Instead of simply modifying an existing bed, Jolien went the extra step and built her own, using just a jig-saw and drill as that's all she had.

▲ The control box uses a simple LCD display for weather and time

"I built a double loft bed in a tree-house style with LED lights and an embedded display with a clock and the weather forecast," she says. "The control panel also has three buttons which can be used to turn the lights on and off. The light strands I bought were connected to battery packs, so I cut those off and wired the strings to a Raspberry

▶ While the top bunk is the more interesting part, the bottom bunk is also great

> ❝ Living in Amsterdam, more living space in my apartment is always very welcome ❞
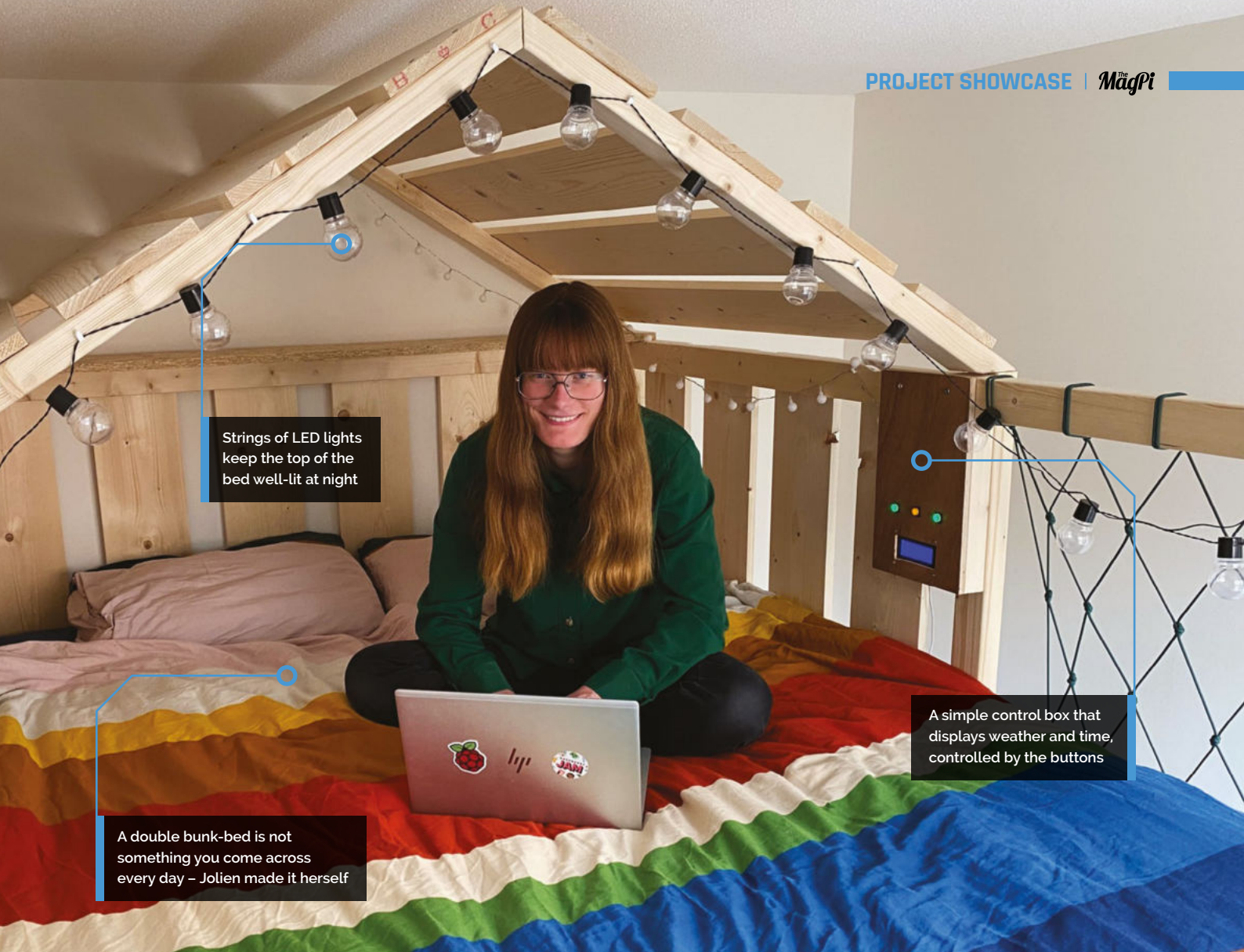
Strings of LED lights keep the top of the bed well-lit at night

A simple control box that displays weather and time, controlled by the buttons

A double bunk-bed is not something you come across every day – Jolien made it herself

Quick **FACTS**

> Jolien tied the safety nets herself

> It all runs off a single Raspberry Pi (for now)

> There's approximately six metres of light strips

> The control panel buttons are arcade buttons

> Weather forecasting was a later addition

◄ When we asked Jolien how much wood she used, she merely said 'a lot'

▲ The bunk-bed allows a better use of the space in Jolien's apartment. It also has stairs instead of a ladder

▶ An early sketch for the design, showing the sloped 'roof' and string of lights

⚠
Warning!
Power tools

Please use safety goggles and gloves when drilling, sawing, and sanding

Pi." The bed is approximately 3 m by 1.7 m and its height is 2.6 m, while its steps also function as shelves for the ground floor.

Building and wiring

"When I started building, I wasn't sure what features I wanted to add and Raspberry Pi is very flexible," Jolien recalls. "It's also very easy to set up SSH, which was necessary since the electronics are embedded in the side panel and can't be reached directly."

> 💬 I also made a progressive web app which enables me to turn the lights on/off 💬

For just a bed, it's got some fairly advanced tech in it, as Jolien reveals: "Besides the control panel in the bed, I also made a progressive web app which enables me to turn the lights on/off and set an alarm via my smartphone. If the alarm is triggered, Raspberry Pi turns on the lights and a song is played. Everything runs on Python and I used Python Flask to make an API on Raspberry Pi so that the GPIO pins connected to the lights can be turned HIGH and LOW from the outside."

Build a smart bed

01 Without a frame you don't have a bed. Jolien built this with just a jig-saw and a drill – although those are two very good power tools for making such a thing.
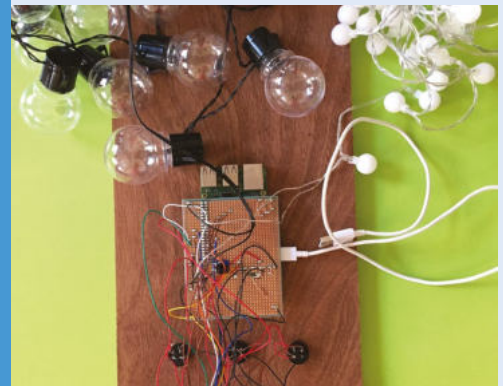
02 The bed isn't smart if there are no Internet of Things capabilities. A Raspberry Pi controls the lights and a small LCD screen for weather and time.

LOFT BED INTERFACE

Good afternoon!

Lights front

Lights back

Rise and shine at
08:33
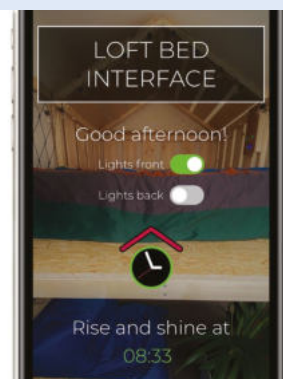
03 To truly make it the smartest bed, you need a smartphone app. Jolien developed a phone app that controls the bed at a touch of a button.

▲ The finished loft bed offers a luxurious sleeping experience

◀ Building the timber frame of the double bunk

Sleep on it

It sounds amazing already, and we'd probably put down some serious cash for a product version of it. However, Jolien isn't quite finished yet. "I think this project took me about 100 hours in total," she says. "At this moment, the speaker is not attached yet; I will do this next week. I also want to design a 3D-printed laptop wall mount for the ground floor."

We're eager to see what else Jolien can come up with to add to an already very impressive project. **M**

SUBSCRIBE TODAY FROM ONLY £5

SAVE UP TO 35%

Subscriber Benefits

▷ FREE Delivery
Get it fast and for FREE

▷ Exclusive Offers
Great gifts, offers, and discounts

▷ Great Savings
Save up to 35% compared to stores

Rolling Monthly Subscription

▷ Low monthly cost (from £5)
▷ Cancel at any time
▷ Free delivery to your door
▷ Available worldwide

Subscribe for 12 Months

£55 (UK)	£90 (USA)
£80 (EU)	£90 (Rest of World)

Free Raspberry Pi Zero W Kit with 12 Month upfront subscription only (no Raspberry Pi Zero W Kit with Rolling Monthly Subscription)

📞 Subscribe by phone: **01293 312193**

➤ Subscribe online: **magpi.cc/subscribe**

✉ Email: **magpi@subscriptionhelpline.co.uk**

JOIN FOR 12 MONTHS AND GET A

FREE Raspberry Pi Zero W Starter Kit

WITH YOUR SUBSCRIPTION

Subscribe in print for 12 months today and you'll receive:

WORTH
£20

- ▷ Raspberry Pi Zero W

- ▷ Raspberry Pi Zero W case with three covers

- ▷ USB and HDMI converter cables

- ▷ Camera Module connector

  Offer subject to change or withdrawal at any time
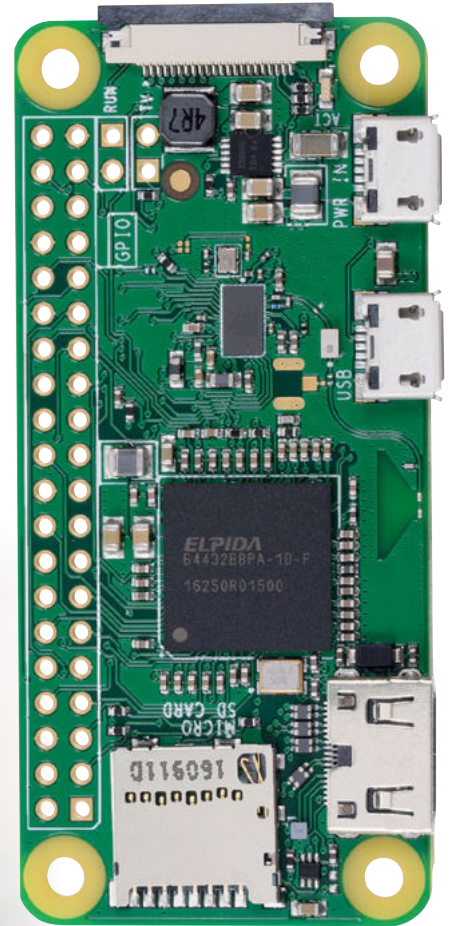
8GB RASPBERRY

40x THE POWER OF THE ORIGINAL RASPBERRY PI!

With twice the memory, Raspberry Pi 4 8GB is the cream of the crop

Raspberry Pi 4 is the fastest Raspberry Pi ever made, and it comes with a choice of memory capacity: 2GB and 4GB. Now there's a new top-end Raspberry Pi 4 8GB. With twice the memory of the top launch model, Raspberry Pi 4 8GB is designed for power users. Whether you're looking to run large machine learning models, speed up multitasking and file access, or work with virtual machines, you'll find a lot more headroom for your workload.
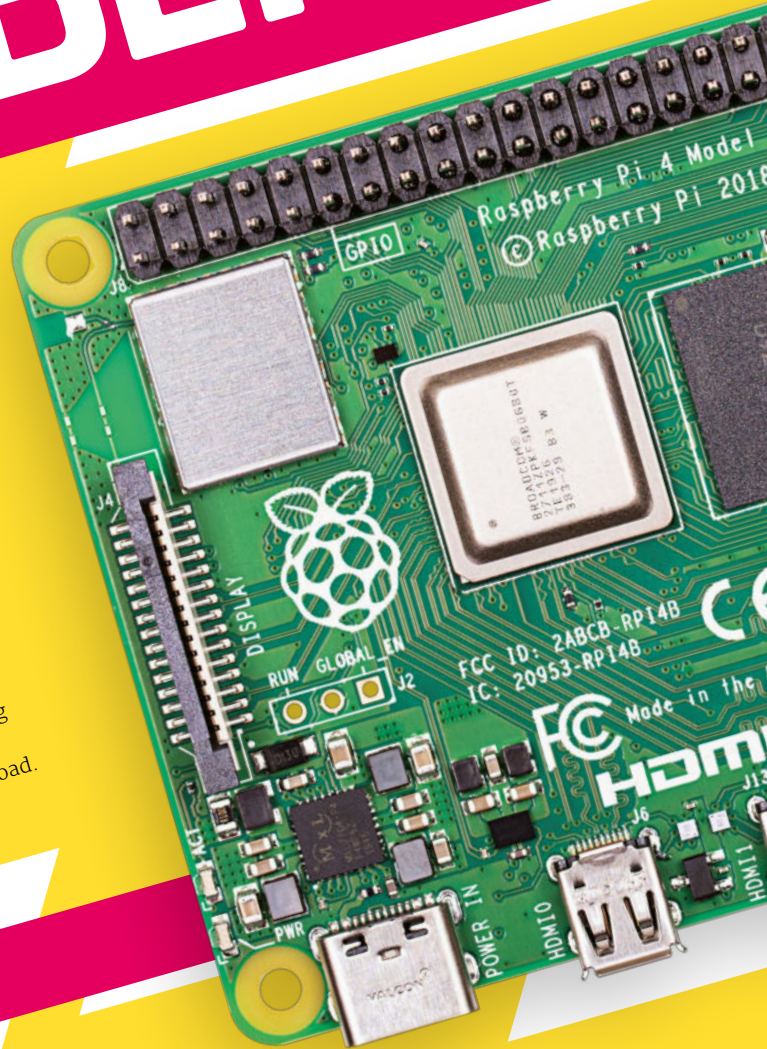
PI 4

PLACES TO BUY

UK & Ireland

Raspberry Pi Store
magpi.cc/retail-store

OKdo
okdo.com

The Pi Hut
thepihut.com

CPC
cpc.farnell.com

Pimoroni
pimoroni.com

North America

OKdo
okdo.com

Adafruit
adafruit.com

Micro Center
microcenter.com

PiShop.us
pishop.us

Newark
newark.com

Canakit
canakit.com

Europe

OKdo
okdo.com

Kiwi Electronics
kiwi-electronics.nl

Kubii
kubii.fr

Melopero
melopero.com

pi3g
pi3g.com

BuyZero
buyzero.de

Sertronic
digitec.ch

SEMAF
electronics.semaf.at

Totonic
pi-shop.ch

Electrokit
electrokit.com

Jkollerup
raspberrypi.dk

For a full list of approved resellers, please go to **rpf.io/products**

MEET
RASPBERRY PI 4 8GB

New revision Raspberry Pi 4 board has twice the RAM. Get to know what makes up a Raspberry Pi

SPECIFICATIONS

SoC: Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit @ 1.5GHz

RAM: 2GB, 4GB, or 8GB LPDDR4 SDRAM

GPU: Broadcom VideoCore VI

NETWORKING: 2.4GHz and 5GHz 802.11b/g/n/ac wireless LAN

BLUETOOTH: Bluetooth 5.0, Bluetooth Low Energy (BLE)

GPIO: 40-pin GPIO header, populated

STORAGE: microSD

PORTS: 2 × micro-HDMI 2.0, 3.5 mm analogue audio-video jack, 2 × USB 2.0, 2 × USB 3.0, Gigabit Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

DIMENSIONS: 88 mm × 58 mm × 19.5 mm, 46 g

A CPU

This is the Broadcom chip used in Raspberry Pi 4 Model B. It continues the quad-core CPU design of the BCM2837, but uses the more powerful ARM A72 core. It has a greatly improved GPU feature set with much faster input/output, due to the incorporation of a PCIe link that connects the USB 2.0 and USB 3.0 ports, and a natively attached Ethernet controller. Read more: **magpi.cc/bcm2711**

B POWER

The move to a USB Type-C connector for power allows Raspberry Pi 4 to support higher-current USB devices

C RAM

The BCM2837 SoC is capable of addressing more memory than the SoCs used in Raspberry PI Model 3 (and earlier models). The move to 8GB of LPDDR4 memory, from the maximum of 4GB of previous designs, increases Raspberry Pi 4 performance further

D ETHERNET

The Ethernet port, relocated to the top-right of the board, offers full-speed network connectivity with no bottlenecks
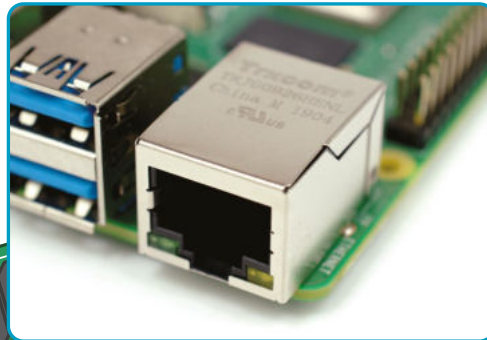
E DUAL DISPLAYS

The two micro-HDMI connectors enable Raspberry Pi 4 to drive two 4K displays at up to 4Kp30, or a single display at up to 4Kp60
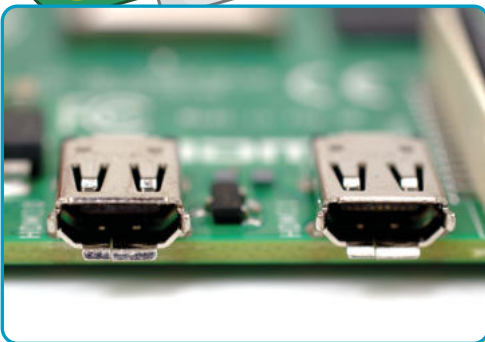
F USB

Two USB 3.0 ports, shaded blue, offer high-speed connectivity for external devices, including storage and accelerator hardware

MAKING THE
MOST OF 8GB

Get the most from your Raspberry Pi's extra RAM with these ideas

Whether you need – or even benefit from – going from 2GB to 4GB or 8GB of memory depends heavily on what you're doing with Raspberry Pi. Here are some examples of workloads that really take advantage of Raspberry Pi 4 8GB's extra memory to improve performance.

RAM disk
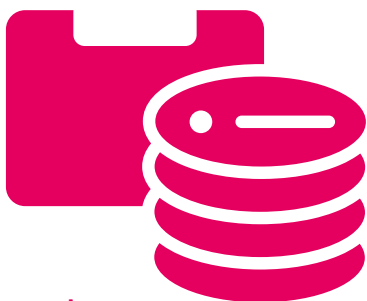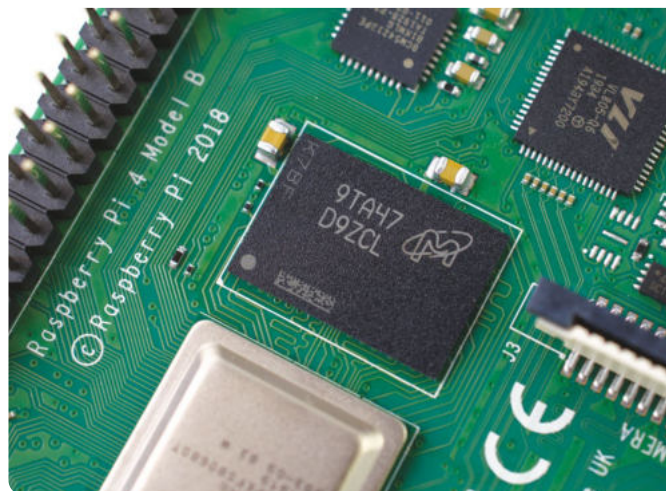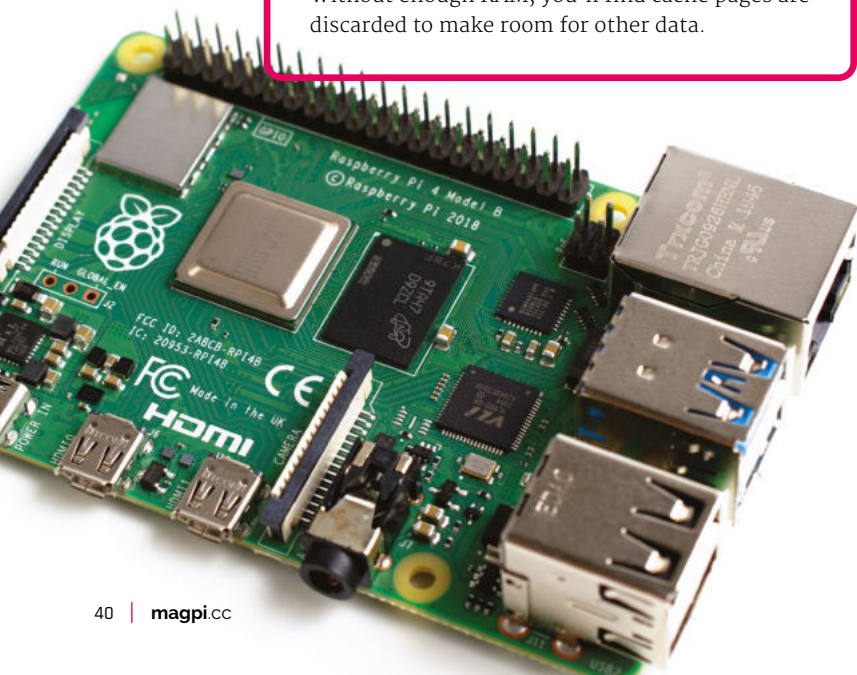Like storage caching, a RAM disk speeds up data access – but under manual control. A RAM disk is orders of magnitude faster than even a USB SSD – but its contents, unless manually saved elsewhere, are discarded when you reboot. A 4GB RAM disk would still leave 4GB free for applications.

Storage cache
The most broadly applicable advantage of having more memory is in accelerating storage access: simply use Raspberry Pi 4 8GB long enough and all your commonly accessed data will be cached in RAM while still leaving room for applications. Without enough RAM, you'll find cache pages are discarded to make room for other data.
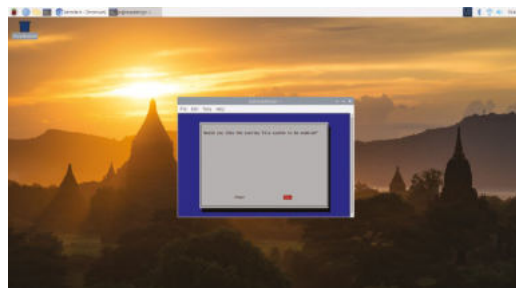
Storage-free computing
The ultimate in performance, storage-free computing sees the entire operating system loaded into RAM – meaning no local storage device is required once booted. Like a RAM disk, this dramatically improves responsiveness – and all changes are lost on reboot. The overlayfs option in raspi-config converts a Raspbian system to be read-only and only ever write changes into a memory storage. It's a great choice for kiosk systems and other public installations.

Image and video processing

Raspberry Pi 4 can speed up image and video processing tasks by running four threads – but only if there's enough RAM. A Guetzli image-compression batch processing job that took over 13 hours on Raspberry Pi 4 2GB finished in two and a half with 4GB – and under one and a half with 8GB.

In-memory databases

For web and database servers, accessing storage is to be avoided at all costs: if you have more than a handful of simultaneous users, performance suffers. With 8GB of RAM, it's possible to move databases from storage into memory – allowing a Raspberry Pi 4 to support lots of users at the same time.
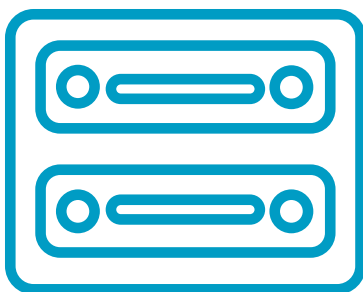
Virtual machines and containers

Containers and virtual machines make it easy for a single computer to perform multiple tasks, acting as though it's multiple distinct computers. They can, however, be memory-hungry – but with Raspberry Pi 4 8GB, you have the RAM to run seven virtual machines with 1GB of RAM each and still have 1GB left over.

Machine learning

Machine learning and artificial intelligence workloads can vary dramatically in how much memory they require: some can run on microcontrollers costing mere pennies apiece, while others need a data centre filled with expensive hardware. Having double the RAM brings larger examples of these workloads to Raspberry Pi 4.

Dual-head; dual-user workstations

It's possible to use any model of Raspberry Pi 4 as a dual-head workstation – giving two users a keyboard, mouse, and monitor of their own. Previously, this meant sharing 4GB of RAM – but with Raspberry Pi 4 8GB, they have the equivalent of a Raspberry Pi 4 4GB each for considerably less money.

EBEN UPTON ON THE UPGRADE TO 8GB

It took memory technology time to catch up with the team's vision for the 8GB model to become a reality

We wanted to be able to do an 8GB model," Eben Upton, founder of the Raspberry Pi Foundation, recalls of Raspberry Pi 4's launch last year, "but there was no 8GB RAM package then."

The result was that Raspberry Pi 4 launched in three models, identical save for the amount of memory on board: an entry-level 1GB, a mainstream 2GB, and a top-end 4GB aimed at power users. Since then, the 2GB model has become the entry-level option.

Now, Raspberry Pi 4 8GB offers an even higher-end option – and it came as quickly as it could.

Waiting on the technology

"We're launching this absolutely as soon as we can," Eben says. "There's no synthetic delay in this product. If we could have launched this on day one we would have done, but the technology just wasn't there yet.

"The 8GB, of course, is about wanting to provide a platform for power users, and wanting to do that just as soon as the technology is ready. You know,

THE MARCH TO 64-BIT

"I think you've seen it on the PC platform where people were able to keep going with 32-bit a little bit longer, with 32-bit plus PAE [Physical Address Extension, a technique for addressing more than 4GB of memory on a 32-bit system], than you would expect, but quite quickly you stop seeing benefits," says Eben.

"The other thing that's pushing us to 64-bit is there are pieces of software that we rely on that are going to start deprecating or just discontinuing support – like Chromium. It's probably not that long until Chromium stops supporting 32-bit.

"Obviously, we're going to keep supporting 32-bit as there's a lot of hardware in the field and we still have Raspberry Pi Zero, so we still have a first-class product which is 32-bit only. But yeah, I think 64-bit is the future."
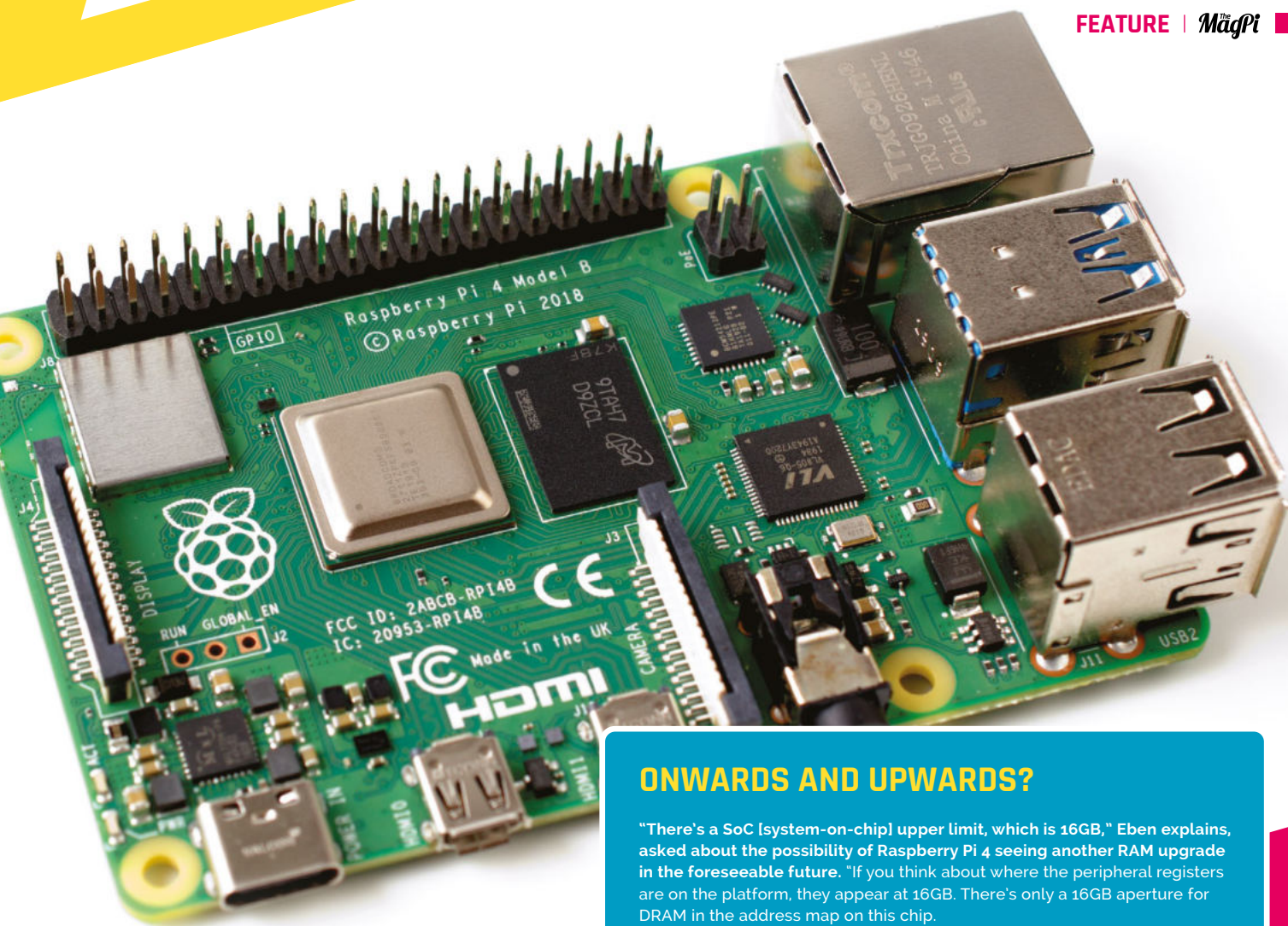
ONWARDS AND UPWARDS?

"There's a SoC [system-on-chip] upper limit, which is 16GB," Eben explains, asked about the possibility of Raspberry Pi 4 seeing another RAM upgrade in the foreseeable future. "If you think about where the peripheral registers are on the platform, they appear at 16GB. There's only a 16GB aperture for DRAM in the address map on this chip.

"Now, there really isn't a 16GB RAM package that we could use, or a way to use a pair of 8GB packages on the board. So, I think this is probably the pinnacle for Raspberry Pi 4 in terms of memory capacity."

when we launched last year this 8GB package didn't exist – but we sort of suspected it might come to exist.

"The memory packages we're using are literally some of the first off the production line. This really is a brand-new, shiny memory technology."

❝ This really is a brand-new, shiny memory technology ❞

Turbines to power

"It's a subset of users that are going to want this 8GB model," Eben predicts, "and it's going to be the power users. I think it's going to people with big data sets, people who want to do database stuff.

"I think it's honestly a nice little server – I mean those Raspberry Pi computers that Mythic Beasts have in the cloud, they're very, very popular. One of the reasons they've been able to serve our website so successfully, on a relatively small budget, is because they have this philosophy where their x86 machines are configured with enormous amounts of RAM and

with battery-backed RAID cards, so effectively your entire database ends up in RAM.

"Reads from the database happen from RAM; writes to the database are posted to this cache on the front-end of the RAID card and aren't synchronous with the drive.

"In this kind of world where you try to make sure it doesn't matter what kind of storage it is – if it's an SD card or spinning rust or an SSD – to some extent, regardless of whether you're Raspberry Pi or a big x86 machine, if you ever touch the backing store synchronously you die, in terms of performance."

"Raspberry Pi 4 8GB is going to be great for people who want lots of data to end up in, and stay in, RAM." 𝗠

THE OFFICIAL **RASPBERRY PI**
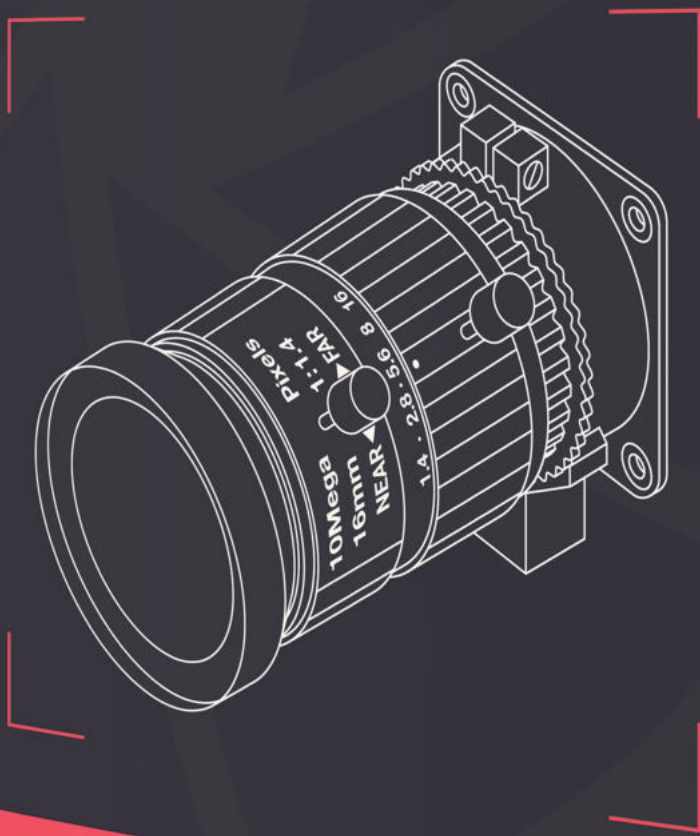
CAMERA GUIDE

FOR CAMERA MODULE & HIGH QUALITY CAMERA

High Quality Camera:
Precise control and time-lapses

Part 02

Explore the many camera options and shoot a time-lapse video

n the first part of this series (see issue 93 **magpi.cc/93**), we showed you how to connect the High Quality Camera or standard Camera Module to your Raspberry Pi and we learned how to take still photos and shoot videos from the command line. Now let's explore the `raspistill` and `raspivid` commands further, including the many switches and options available. We'll also reveal how to shoot a time-lapse video.

Precise camera control

01 **Preview mode**
When taking stills or shooting video, one of the first things you might want to alter is the preview window that appears by default on the screen. First of all, if it's upside-down, just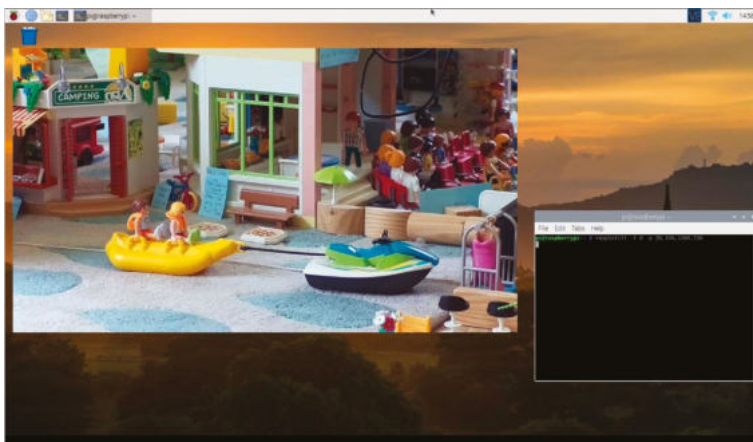 add `-rot 180` to your `raspistill` or `raspivid` command to rotate it. Also, adding `-hf` and/or `-vf` will flip the image horizontally and/or vertically.

▲ The preview can be resized and positioned manually, and also have its opacity adjusted

Using the `-p` switch, you can set the window's on-screen position, along with its height and width. The `-p` switch takes four parameters: x co-ordinate, y co-ordinate, width, and height. So, for example:

```
raspistill -o image.jpg -p 20,100,1280,720
```

…would place the preview window's top-left corner at co-ordinate (20,100), with a width of 1280 pixels and height of 720 pixels.

Note that if you only want to see a preview without taking a shot, you can simply omit the `-o image.jpg` part. The `-t` switch sets the duration of the preview: you can set it to 0 to make it stay on screen until you press **CTRL+C**.

If you want a full-screen preview, this is easily achieved using the `-f` switch. The `-op` switch can be used to adjust the preview's opacity, from 0 (invisible) to 255 (solid). If you want to disable the preview window completely, use the `-n` switch.

02 **Camera control options**
Like most dedicated digital cameras, the Raspberry Pi HQ Camera or Camera Module offers a range of options to adjust aspects such as brightness (`-br`, from 1 to 100), contrast (`-co`, -100 to 100), sharpness (`-sh`, -100 to 100), saturation (`-sa`, -100 to 100), ISO (`-ISO`, 100 to 800), and EV compensation (`-ev`, -10 to 10). In addition, there are numerous options for exposure mode for shooting in certain scenarios, akin to the 'scenes' found on most digital cameras. Just use the `-ex` switch followed by one of the following terms: `auto`, `night`, `nightpreview`, `backlight`, `spotlight`, `sports`, `snow`, `beach`, `verylong` (long exposure), `fixedfps` (for video only), `antishake`, or `fireworks`.

▲ A multitude of real-time effects may be added to images, including emboss, as shown here

Automatic white balance can be adjusted by following the `-awb` switch with one of: `off`, `auto`, `sun`, `cloud`, `shade`, `tungsten`, `fluorescent`, `incandescent`, `flash`, or `horizon`.

You can set the shutter speed in microseconds with the `-ss` switch; the upper limit depends on the exposure mode and other settings. The metering mode – used for preview and capture – can be set with `-mm` to one of the following: `average`, `spot`, `backlit`, or `matrix`.

There's also the option of restricting the region of interest (ROI) to only part of the sensor, using `-roi` with parameters for x and y co-ordinates (from top left), width, and height. For example, to set an ROI halfway across and down the sensor, with quarter-size width and height, you'd use: `-roi 0.5,0.5,0.25,0.25`.

> ❝ You can set the shutter speed in microseconds with the -ss switch ❞

03 Keypress mode

If you'd like to take a still photo at an exact time, rather than having to wait for the `-t` switch delay time to elapse, keypress mode is your friend. Just add the `-k` switch to your `raspistill` command, then press the **ENTER** key to take the shot: it acts like a shutter button. To exit the procedure, press **X** followed by **ENTER**.

By adding `%04d` to the end of your file name in the command, you can save every shot you have taken before aborting:

```
raspistill -o keypress%04d.jpg -k
```

Each shot will have a four-digit sequential number added to its file name – so, **keypress0000.jpg**, **keypress0001.jpg**, **keypress0002.jpg**, etc. This is a useful technique for time-lapses using the `-tl` switch, too, which we'll explore later.

04 Image effects

A whole bunch of effects can be added to the camera in real-time, shown in the preview window. This is achieved by using the `-ifx` switch followed by one of the following terms: `none`, `negative`, `solarise`, `posterise`, `sketch`, `denoise`, `emboss`, `oilpaint`, `hatch`, `gpen` (graphite sketch effect), `pastel`, `watercolour`, `film`, `blur`, `saturation` (adjust colour saturation of the image), `colorswap`, `washedout`, `colorpoint`, `colorbalance`, or `cartoon`.

If you'd like to take monochrome images, you can use the `-cfx` (colour effect) switch to achieve this, using the following setting: `-cfx 128:128`.

To increase contrast between dark and light areas using DRC (dynamic range compression), use the `-drc` switch to turn it on/off (it's off by default).

05 Still options

Let's take a look at some options that are specific to the `raspistill` command. As already mentioned, we use `-o` followed by a file name to output to a file, and the `-t` switch sets the shutter delay in milliseconds. For example, to save a photo taken after two seconds, use:
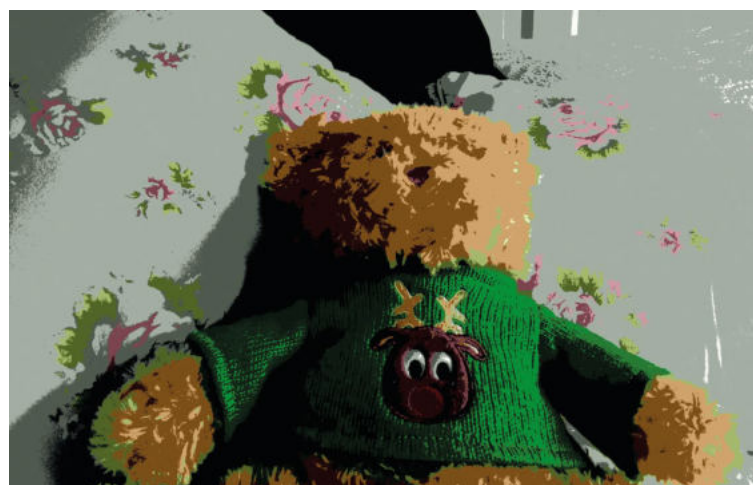
```
raspistill -t 2000 -o image.jpg
```


▼ The posterise effect is shown here; use `-ifx posterise` in your command

Top Tip 👍

Other video formats

WebM is an open video format that can be displayed directly in most browsers. However, other video formats are available.

You can set the width and height of the image with -w and -h, each followed by a value – up to 4056 and 3040 for the HQ Camera (or 3280 and 2464 for Camera Module v2; 2592 and 1944 for Camera Module v1).

You can also set the quality of the JPEG image, using -q, from 0 to 100 – the latter is almost completely uncompressed. Alternatively, to save it as a lossless PNG (slower than using JPG), use -e (encoding) followed by png:

```
raspistill -o image.png –e png
```

The raspiyuv command works in a similar fashion and offers most of the same options, apart from adding EXIF tags, but sends its YUV or RGB output directly from the camera component to file. To use RGB, add the -rgb switch.

06 Shooting video

The raspivid command is used to shoot video. In this case, the -t switch sets the duration in milliseconds. The bitrate is set using -b, in bits per second (so, 25Mpbs is -b 25000000), while -fps sets the frame rate. For example, to shoot five seconds of video at 1080p (1920 × 1080), with a bitrate of 15Mbps and frame rate of 30fps, use:

```
raspivid -t 5000 -b 15000000 -fps 30 -o
video.h264
```

Many other video options are available, including time delays, keypress mode, and segmenting a stream into multiple files.

Time-lapse photography

▼ Running the rendering process on a Raspberry Pi. This will take some time, so you may prefer to use a faster machine

01 Speeding it up

Time-lapse photography reveals exciting things about the world which you wouldn't otherwise be able see. These are things that

happen too slowly for us to perceive: bread rising and plants growing; the clouds, sun, moon, and stars crossing the sky; shadows moving across the land. We'll be making a Raspbian-based device that lets you watch things that are too slow to observe with the naked eye. To do this, we will capture lots of still photos and use FFmpeg to combine these frames into a video which can then be accessed via a web browser.

> ❝ We will capture lots of still photos and combine these frames into a video ❞

02 Install and configure software

Install a web server so you can access your captured images remotely. Run this command to install Apache:

```
sudo apt-get install apache2
```

Remove the default page to see the contents of the directory:

```
sudo rm /var/www/html/index.html
```

Visit the IP address of your Raspberry Pi (e.g. **http://192.168.1.45**) – which you can find by using hostname -I – on another computer and you should see an empty directory listing. If you run the following command and refresh the page, you should see an image file listed. You need to run this as a superuser (by using sudo) so you can write to the directory.

```
sudo raspistill -o /var/www/html/testimage.jpg
```

Click on the file link on the remote computer and you'll see the image in your browser.

03 Capture the images

Set up your scene and check the positioning of the camera.

```
sudo raspistill -w 1920 -h 1080 -o
/var/www/html/testimageFullHD.jpg
```

The width and height have been changed to capture a smaller image in 16:9 aspect ratio. This

◄ Time-lapse photography is ideal for capturing lengthy natural processes

makes things easier later. The top and bottom are cropped, so make sure that your subject is in frame. Run this to start the capture:

```
sudo raspistill -w 1920 -h 1080 -t 10800000
-tl 10000 -o /var/www/html/frame%04d.jpg &
```

This takes a photo every ten seconds for three hours. The '&' at the end causes the process to run in the background.

04 Prepare to make the video

You can render the video on Raspberry Pi, but – depending on the model – this can take a long time. So you may prefer to transfer the files to a more powerful computer instead. Whichever method you decide to use, you will need to install the tools on the rendering machine; to do so on Raspberry Pi, enter:

▲ Some bread dough ready to prove. Watch it rise in your video. Be careful not to move the bowl or camera during filming

```
sudo apt-get install ffmpeg
```

This installs the FFmpeg tool which we'll use to convert our images into a video. To copy the images to a remote machine, you can download them from the web server using wget or curl. For example:

```
wget -r -A jpg http://192.168.1.45
```

Or, if you don't have wget…

```
curl http://192.168.1.45/frame [0001-0766].jpg -O
```

Change the IP address and numbers accordingly.

05 Make the video

The final step is to assemble the video. Run the following command to start the rendering process:

```
sudo ffmpeg -i /var/www/html/frame%04d.jpg
-crf 4 -b:v 10M /var/www/html/video.webm
```

When the rendering process has finished, you'll be able to view the video in your browser. The default frame rate is 25fps. This compresses three hours of images taken at ten-second intervals to about 40 seconds of video. You can adjust this with the `-framerate` command-line option. Note that in our example, the bitrate (`-b`) has been set high, and the Constant Rate Factor (`-crf`) has been kept low, to produce a good-quality video. M
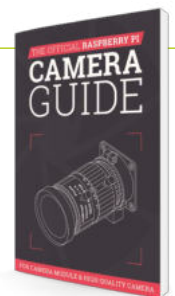
The Official Raspberry Pi Camera Guide

For further tutorials on how to use your HQ Camera or Camera Module, take a look at our new book, *The Official Raspberry Pi Camera Guide*. Its 132 pages are packed with essential info and a range of exciting projects. **magpi.cc/cameraguide**

Make a 3D camera

Add some depth to your photography by taking true 3D pictures with the new High Quality Camera

![PJ Evans photo]

MAKER

PJ Evans

PJ is a writer, software engineer, 3D photography enthusiast, and organiser of the Milton Keynes Raspberry Jam.

@mrpjevans

With the release of the High Quality Camera, photography using Raspberry Pi has been taken to a whole new level. A 12.3 megapixel sensor and support for multiple lenses has already had enthusiasts experimenting with new types of photography previously out of their reach. Adding a Raspberry Pi to the mix allows for many possibilities, such as time-lapses or connection to sensors. Here, we're going to use two cameras together to create amazing 3D photos that can be viewed in a variety of ways, even without glasses.

You'll Need

- 2 × Raspberry Pi Zero W
  magpi.cc/pizerow

- 2 × High Quality Camera
  magpi.cc/hqcamera

- 2 × 6 mm or 16 mm lens
  magpi.cc/hqlens

- 2 × Official Raspberry Pi Zero Case
  magpi.cc/pizerocase

- Google Cardboard (optional)
  magpi.cc/cardboard

- 3D-printed mounting plate (optional)
  magpi.cc/hq3dmount

- 8 × 2.5 M 6 mm bolts and nuts (optional)

01 Prepare Raspberry Pi

To create 3D photographs, we need to be able to take two photos simultaneously, about 5 cm apart. These images can then be processed into a variety of different formats such as parallel view, cross-view, or anaglyph (when you wear red/green glasses). As this is a 3D project, it's not surprising to learn that we'll need two of everything. As we can only attach one camera per Raspberry Pi Zero W, we'll need to prepare a left and right computer. Start by installing Raspbian Lite as normal on both computers and updating everything with `sudo apt update && sudo apt upgrade`. Make sure both computers are connected to WiFi before proceeding.
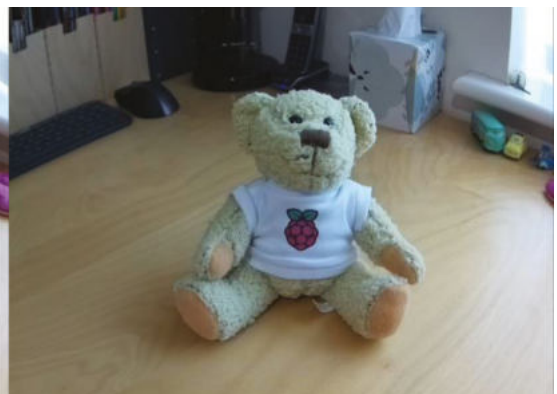
02 Know the difference

Choose one computer for the left camera and the other for the right. From the command line, run `sudo raspi-config` and go to Networking Options > Hostname. Change the name from 'raspberrypi' to 'leftcam' and 'rightcam' on each respective Raspberry Pi Zero. Also in raspi-config, make sure SSH is enabled on each (Preferences > Interfaces).

After this, leave the configuration utility and shut both computers down. Now is a very good time to attach the short camera cable that was supplied with the case to each Raspberry Pi Zero and thread it through the slot on the rear of the case and insert both computers. Add the cover and label each case as 'left' or 'right'.

03 Power sharing

As we have two Raspberry Pi Zero boards, we need two power supplies, right? Well, we can pull a little trick so that only one is required. With some wire, solder a 5 V line and a ground (GND) line from one GPIO to the equivalent on the other (see **Figure 1** diagram). This 'power rail' allows the second Raspberry Pi Zero W to pull power from the one connected to a USB power supply. Just remember to use a suitable power supply with enough amperes to power both. We found

▲ Here's an example of what can be produced. This is a cross-view image, so, if you can, cross your eyes together until they settle on a central image. Don't strain if your eyes feel tired!

Raspberry Pi Zero W only supports a single camera connection, so two are needed

Two 12.3 megapixel cameras provide glorious high-quality images, slightly apart to create the 3D effect

the official micro USB supply worked well. Check, check, and double-check before following this step. Soldering to the wrong connectors could permanently damage your devices.

04 Attach the cameras

It's really important that the two cameras are lined up together and not at odd angles. We've provided a STL file for 3D-printing a mounting plate that holds them perfectly in place. You don't have to use it, but if you do have access to a 3D printer, it'll make life easier. Connect the short ribbon cables provided with the cases to each camera, then attach the cameras to the mounting plate side-by-side using the nuts and bolts. Be very careful not to bend or tear the ribbon cables. Finally, flip the cameras over so the plate is resting on the case lids and affix them with some sticky pads.

05 Camera testing

Before going any further, test that both cameras are working as expected. Carefully attach the lenses to each camera board (if you're using the mounting plate, watch out for the control levers hitting each other). Use SSH to log in remotely to your left camera (`ssh pi@leftcam.local`) and at the command line, enter this:

Figure 1
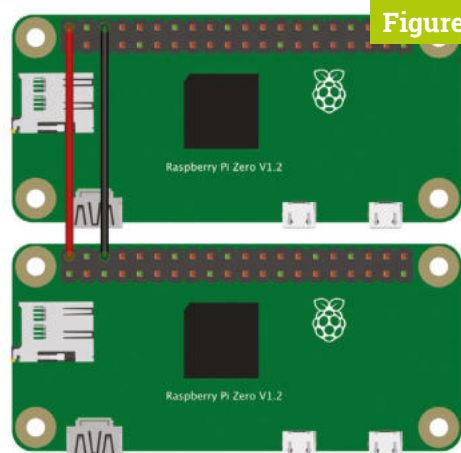
```
raspistill -o test.jpg
```

After a few seconds, an image file will be created in your current directory. Transfer it back to your computer and have a look. Chances are it'll be blurry – but so long as an image was taken, we're all good. Repeat this test on the right-hand camera.

06 Streaming for two

If you've been wondering how one camera is going to get its image to the other, the answer is by setting up a HTTP-based stream on each

Figure 1 Here, a 5V GPIO pin and GND pin are connected to their equivalents on the other GPIO so one Raspberry Pi Zero W provides power for the other

Top Tip 👍

Parallel

Parallel (left/right) is needed for devices like Google Cardboard. Without it, you can get the effect by looking into the distance until the image merges.

camera. This will turn each Raspberry Pi Zero W into a streaming webcam and we can then view both images from a further website we'll install later. The following steps need to be followed on each Raspberry Pi Zero W. Start by installing some libraries we need:

```
sudo apt install cmake libjpeg8-dev git
```

Now we'll get and build the MJPEG streaming software:

```
git clone https://github.com/jacksonliam/
mjpg-streamer.git
  cd mjpeg-streamer/mjpeg-streamer-
experimental
  make
  sudo make install
```

07 First test images

To run the package we've just installed, enter the following commands on each Raspberry Pi Zero W:

```
cd ~/mjpeg-streamer/mjpeg-streamer-
experimental
  export LD_LIBRARY_PATH=.
  ./mjpg_streamer -o "output_http.so -w ./www"
-i "input_raspicam.so"
```

▼ So we're not troubled by multiple power supplies, we can power one Raspberry Pi Zero from the other

There is now a web server running on each device. Have a look by visiting **http://leftcam.local:8080/** and **http://rightcam.local:8080/**. Each will have a fun little website where you can view static and

video feeds from each camera. When you're done, you can stop each server by entering **CTRL+C** in the Terminal.

08 Bring them together

To view both images on the same page and generate 3D images, we need a further web service that takes a feed from both sites we've just installed. Create a directory called **3dcamera** in your home directory, then create two files: **3dcamera.py** and **control.html**. Enter both code listings (or download them from **magpi.cc/3dcameragit**) and save. This is a very simple web server and an HTML page that will display both images on a single page and, with a simple click, create and download a parallel-eye image.

Make sure both MJPEG streamers are running and then start the additional server on leftcam only:

```
python3 3dcamera.py
```

You should be able to access the site at **http://leftcam.local:8081/** and be able to see a video stream of each device.

09 View your image

Take an image by holding the cameras steady and clicking 'Snap!' on the website. The dual image will be downloaded to your computer. If you have a Google Cardboard kit or one of the widely available mobile phone VR headsets, transfer your parallel image to your mobile phone and then view it in glorious three dimensions. If not, many people can see the image by focusing their eyes 'beyond' the two images until they merge into one. If you're struggling with this, reverse the two captured images as noted in the code to create a cross-view image.

10 Take it further

This is just a starting point for your adventures in 3D photography. We haven't touched on creating anaglyphs or streaming 3D video. Check out the GitHub repo (**magpi.cc/3dcameragit**) for a more advanced version that allows you to set the type of image to generate and adds a few more features. What's the most creative thing you can do with your 3D camera? M
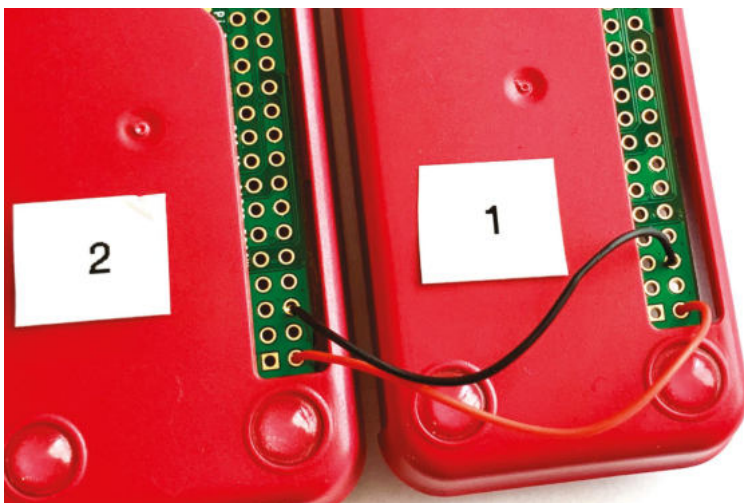
3dcamera.py

> Language: **Python 3**

```python
001.  import os
002.  import urllib.request
003.  from http.server import SimpleHTTPRequestHandler,
      HTTPServer
004.  from PIL import Image
005.  from io import BytesIO
006.
007.  port = 8081
008.  control_html = os.path.dirname(os.path.realpath(
      __file__)) + '/control.html'
009.  # Reverse the URLs to create cross-view images instead
      of parallel
010.  left_camera =
      "http://leftcam.local:8080/?action=snapshot"
011.  right_camera =
      "http://rightcam.local:8080/?action=snapshot"
012.
013.
014.  def process_image():
015.      left_image = urllib.request.urlopen(left_camera)
016.      right_image = urllib.request.urlopen(right_camera)
017.      images = [Image.open(x) for x in [left_image,
      right_image]]
018.
019.      widths, heights = zip(*(i.size for i in images))
020.      total_width = sum(widths)
021.      max_height = max(heights)
022.
023.      side_by_side_image = Image.new('RGB', (
      total_width, max_height))
024.
025.      x_offset = 0
026.      for image in images:
027.          side_by_side_image.paste(image, (x_offset, 0))
028.          x_offset += image.size[0]
029.
030.      image_buffer = BytesIO()
031.      side_by_side_image.save(image_buffer,
      format='JPEG')
032.      image_data = image_buffer.getvalue()
033.
034.      return image_data
035.
036.
037.  class handle_request(SimpleHTTPRequestHandler):
038.      def do_GET(self):
039.          print('Sending control HTML')
040.          f = open(control_html, 'rb')
041.          self.send_response(200)
042.          self.send_header('Content-type', 'text/html')
043.          self.end_headers()
044.          self.copyfile(f, self.wfile)
045.
046.      def do_POST(self):
047.          new_image = process_image()
048.          self.send_response(200)
049.          self.send_header('Content-type', 'image/jpeg')
050.          self.send_header('Content-disposition',
051.                          'attachment; filename="3d.
      jpg"')
052.          self.end_headers()
053.          self.wfile.write(new_image)
054.
055.
056.  httpd = HTTPServer(("", port), handle_request)
057.  try:
058.      httpd.serve_forever()
059.  except KeyboardInterrupt:
060.      pass
061.
062.  httpd.server_close()
```

control.html

> Language: **HTML**

```html
001.  <!DOCTYPE html>
002.  <html>
003.    <head>
004.      <style>
005.        .viewfinder {
006.          display: flex;
007.        }
008.      </style>
009.    </head>
010.    <body>
011.      <h1>3D Camera</h1>
012.      <div class="viewfinder">
013.        <img src="http://leftcam.
      local:8080/?action=stream" />
014.        <img src="http://rightcam.
      local:8080/?action=stream" />
015.      </div>
016.      <form method="post" action="/">
017.        <button>Snap!</button>
018.      </form>
019.    </body>
020.  </html>
```

MAKER

Frank Delporte

Java developer at work, sharing knowledge as a blogger and CoderDojo lead coach. Programming since he was eleven and got a Commodore 64.

webtechie.be

Java electronics: build a Maven app

Part 02

Let's build a full Java Maven application on the Raspberry Pi to control some basic components

I n *The MagPi #93*, we programmed our first Java code on the Raspberry Pi and installed Visual Studio Code. Now it's time for the next step! We will extend the basic code we've already created to a full-blown compiled application using the Pi4J dependency.

01 Getting started

Re-read the article on page 50 in *The MagPi #93* (**magpi.cc/93**) to prepare the Raspberry Pi and microSD card. Install Visual Studio Code and connect the LED and button.

02 Maven

Maven is a tool to help you to manage the dependencies (external libraries) of your Java project and build the application for testing and distribution. You'll need to install it first on your Raspberry Pi and then you can request the version to test it.

```
sudo apt install maven
mvn -v

Apache Maven 3.6.0
Maven home: /usr/share/maven
Java version: 11.0.6, vendor: Raspbian,
runtime: /usr/lib/jvm/java-11-openjdk-armhf
 Default locale: en_US, platform encoding:
UTF-8
 OS name: "linux", version: "4.19.97-v7l+",
arch: "arm", family: "unix"
```

03 Pi4J framework

The easiest way to work with the GPIO pins is by using the Pi4J library in Java. Under the hood, it uses WiringPi to talk to the hardware, handle input changes etc.

You'll also need to install the Pi4J framework on your Raspberry Pi and upgrade WiringPi to the latest version 2.52 (if you are using a Raspberry Pi 4).

```
curl -sSL https://pi4j.com/install | sudo
bash
cd /tmp
wget https://project-downloads.drogon.net/
wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v

gpio version: 2.52
```

Now we are all set to dive into real Java programming with hardware components!

> **Maven helps you to manage the dependencies of your Java project**

04 Start a new project

Let's put Maven to work to create an empty project for us as a starting point, by following these steps. In the our home directory we tell Maven (mvn) to generate a project, 'myFirstApp'.

```
cd /home/pi
mvn archetype:generate -DgroupId=magpi.
app -DartifactId=myFirstApp
-DarchetypeArtifactId=maven-archetype-
quickstart -DarchetypeVersion=1.4
-DinteractiveMode=false
```
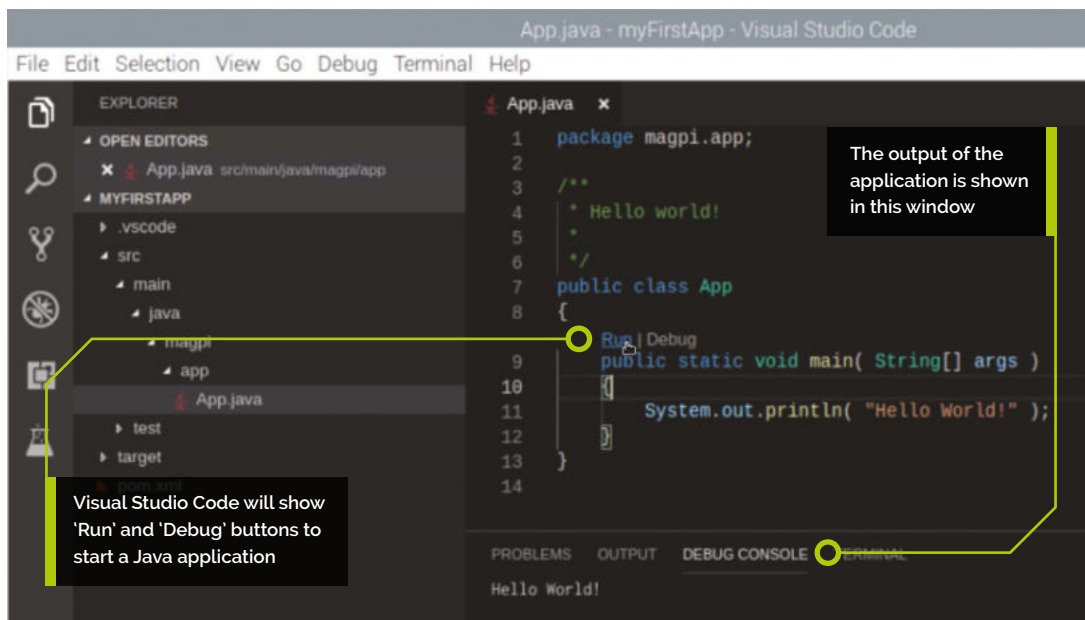
Now we can check what has been created.

```
ls -l myFirstApp

total 8
-rw-r--r-- 1 pi pi 2653 May  7 13:53 pom.xml
drwxr-xr-x 4 pi pi 4096 May  7 13:53 src
```

You'll Need

> Raspberry Pi 2, 3, 3A+, 3B+, or 4

> Raspbian Full version

> LED

> 2 × 330 Ω resistor, push-button, breadboard, and wires

> Optional T-Cobbler connection between Raspberry Pi and breadboard

The output of the application is shown in this window

Visual Studio Code will show 'Run' and 'Debug' buttons to start a Java application

05 Visual Studio Code

Start Visual Studio Code and open the folder **/home/pi/myFirstApp** we just created. In the bottom of the IDE you'll see 'Starting Java Language Server'. As soon as this is finished, you are able to hit the Run|Debug pop-up above the main method and see the result in the output window. Nice! Not a single line of code written by ourselves, but we have a full running Java project!

06 Pi4J dependency

The heart of a Maven project is the **pom.xml** file located in the root directory of the project. Open it in VSC and you will be asked to install an XML extension to help you edit this kind of file.

External libraries you want to use in your project need to be added to the 'dependencies' part. We want to use Pi4J, so we go to **mvnrepository.com** to search for it. We only need to copy the given dependency to our **pom.xml** file.

▲ In the bottom left of VSC, you can see 'Starting Java Language Server' when a Java project is opened

```
<dependency>
<groupId>com.pi4j</groupId>
<artifactId>pi4j-core</artifactId>
<version>1.2</version>
</dependency>
```

From now on, Maven will make sure this dependency is downloaded and added to our compiled application. In VSC we will also get code completion on the methods we use from this library.

07 GpioHelper code

The **GpioHelper.java** code (overleaf) provides us all the functionality we need for our application to control a LED and read a button state. We start with the imports of the classes we use from the Pi4J library. Then we define two values for the pins we want to use. Watch out with the numbers as Pi4J internally uses WiringPi, so we need these numbers – go to **pinout.xyz** to easily find the right ones. In the constructor `GpioHelper()` of this class, the controller, pins, and button listener are initialised. Additional methods to toggle the LED, get the number of presses from the button listener, and to shut down the GpioController are also part of this class.

08 ButtonListener code

The ButtonListener implements the Pi4J GpioPinListenerDigital interface needed to add the functionality we want in our application. In this case we count the number of times the button is pressed and provide a getter for this value.

Top Tip 👍

Video step-by-step

To take a look over your author's shoulder while going through these steps, go to **magpi.cc/ javavideo**, where the steps of the article in *The MagPi* #93 and this one are shown.

GpioHelper.java

> Language: **Java**

```
001.   package magpi.app;
002.
003.   import com.pi4j.io.gpio.GpioController;
004.   import com.pi4j.io.gpio.GpioFactory;
005.   import com.pi4j.io.gpio.GpioPinDigitalInput;
006.   import com.pi4j.io.gpio.GpioPinDigitalOutput;
007.   import com.pi4j.io.gpio.Pin;
008.   import com.pi4j.io.gpio.PinPullResistance;
009.   import com.pi4j.io.gpio.PinState;
010.   import com.pi4j.io.gpio.RaspiPin;
011.
012.   public class GpioHelper {
013.       private static final Pin PIN_LED = RaspiPin.GPIO_03;      // BCM 22
014.       private static final Pin PIN_BUTTON = RaspiPin.GPIO_05;   // BCM 24
015.
016.       private GpioController gpioController;
017.
018.       private GpioPinDigitalOutput led = null;
019.       private GpioPinDigitalInput button = null;
020.       private ButtonListener buttonListener = null;
021.
022.       public GpioHelper() {
023.           try {
024.               // Initialize the GPIO controller
025.               this.gpioController = GpioFactory.getInstance();
026.
027.               // Initialize the led pin as a digital output pin with
       initial low state
028.               this.led = gpioController.provisionDigitalOutputPin(
       PIN_LED, "RED", PinState.LOW);
029.               this.led.setShutdownOptions(true, PinState.LOW);
030.
031.               // Initialize the input pin with pull down resistor
032.               this.buttonListener = new ButtonListener();
033.               this.button = gpioController.provisionDigitalInputPin(
       PIN_BUTTON, "Button",
034.                   PinPullResistance.PULL_DOWN);
035.               button.addListener(this.buttonListener);
036.           } catch (UnsatisfiedLinkError | IllegalArgumentException ex) {
037.               System.err.println("Problem with Pi4J!"
038.                   + " Probably running on non-Pi-device or Pi4J not
       installed. Error: "
039.                   + ex.getMessage());
040.           }
041.       }
042.
043.       public void toggleLed() {
044.           if (this.led != null) {
045.               if (this.led.isLow()) {
046.                   this.led.high();
047.               } else {
048.                   this.led.low();
049.               }
050.           }
051.       }
052.
053.       public ButtonListener getButtonListener() {
054.           return this.buttonListener;
055.       }
056.
057.       public void close() {
058.           this.gpioController.shutdown();
059.       }
060.   }
```

```
package magpi.app;

import com.pi4j.io.gpio.PinState;
import com.pi4j.io.gpio.event.
GpioPinDigitalStateChangeEvent;
import com.pi4j.io.gpio.event.
GpioPinListenerDigital;

public class ButtonListener implements
GpioPinListenerDigital {

    private long pressed = 0;

    /**
     * Event handler for the button
     */
    @Override
    public void handleGpioPinDigitalState
ChangeEvent(GpioPinDigitalStateChangeEvent
event) {
        if (event.getState().
equals(PinState.LOW)) {
            this.pressed++;
            System.out.println("Button is
pressed for the " +
                this.pressed + "th time");
        }
    }

    public long getPressed() {
        return this.pressed;
    }
}
```

09 Main code

Because we moved all the GPIO code into its own classes, we can limit the code in the main method. We only need to initialise the GpioHelper and can toggle the LED until the button has been pressed ten times.

```
package magpi.app;

public class App {
    public static void main(String[] args)
throws InterruptedException {
        System.out.println("Started");

        GpioHelper gpioHelper = new
GpioHelper();

        while (gpioHelper.
getButtonListener().getPressed() < 10) {
            gpioHelper.toggleLed();
            Thread.sleep(250);
        }
```

```
        gpioHelper.close();

        System.out.println("Finished");
    }
}
```

Our application is now finished and you can start it with the 'Run' pop-up above the main method.

---

10 **Packaging the application**

In case you develop the application on your PC or want to take it from one Raspberry Pi to another, you'll need to package it to a JAR file. This file will contain the compiled code and dependencies. Change the build section in the **pom.xml** file to use the 'maven-assembly-plugin' which will handle this packaging – see the edited code listing on this page.

Now we can let Maven build and package our app. You'll see a lot of logging of the work Maven is doing, with 'BUILD SUCCESS' at the end.

```
mvn clean package
…
[INFO] -------------------------------------
-------------------------------------
[INFO] BUILD SUCCESS
[INFO] -------------------------------------
-------------------------------------
[INFO] Total time:  14.636 s
[INFO] -------------------------------------
-------------------------------------
```

Once you've done this on your PC, copy the resulting JAR file to your Raspberry Pi via SSH or a USB memory stick.

---

11 **Running the compiled code**

The compiled JAR file can now be used on any Raspberry Pi with Java 11 (or newer) and the Pi4J framework (Step 03). Just copy this one file and start it with `java -jar <FILENAME>`, like this:

```
cd /home/pi/myFirstApp/target
java -jar myFirstApp-1.0-SNAPSHOT-jar-with-
dependencies.jar

Started
Button is pressed for the 1th time
Button is pressed for the 2th time
…
Button is pressed for the 10th time
Finished
```

pom.xml

> Language: **Java**

```
001.  <build>
002.      <plugins>
003.          <plugin>
004.              <artifactId>maven-assembly-plugin</artifactId>
005.              <version>2.2.1</version>
006.              <configuration>
007.                  <descriptorRefs>
008.                      <descriptorRef>jar-with-dependencies</
descriptorRef>
009.                  </descriptorRefs>
010.                  <archive>
011.                      <manifest>
012.                          <addClasspath>true</addClasspath>
013.                          <mainClass>magpi.app.App</mainClass>
014.                      </manifest>
015.                  </archive>
016.              </configuration>
017.              <executions>
018.                  <execution>
019.                      <id>make-assembly</id>
020.                      <phase>package</phase>
021.                      <goals>
022.                          <goal>single</goal>
023.                      </goals>
024.                  </execution>
025.              </executions>
026.          </plugin>
027.      </plugins>
028.  </build>
```
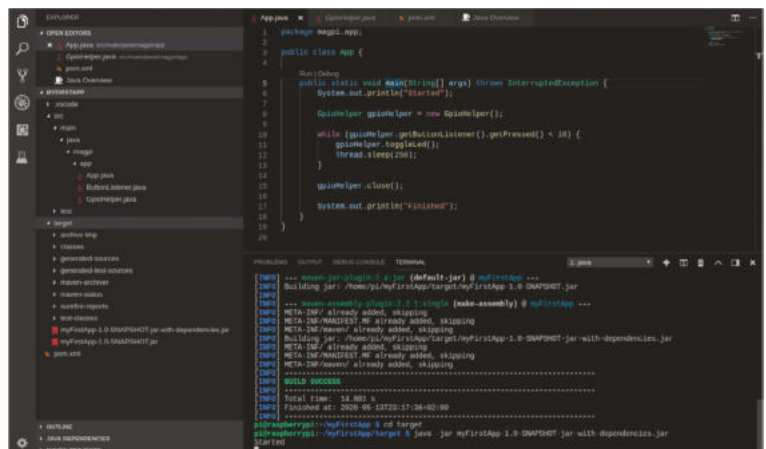

---

12 **A final task**

We haven't added new functionality to the project we started in the previous issue's guide, but we've converted it to a real Java application now! Can you rework this code to count how many times you can push the button in five seconds? Or increase the blink speed related to the number of times the button is pressed? Experiment with the given code so you get a better understanding of how Java works! ⓜ

▲ Using the Terminal window within VSC, we can compile and run our application

Make some noise with Raspberry Pi

If your amazing project is a little too quiet, add high-fidelity sound with Raspberry Pi

PJ Evans

**y** @mrpjevans

PJ Evans is a developer and wrangler of the Milton Keynes Raspberry Jam. He runs a LoRa gateway, which is probably the nearest he'll get to his own radio breakfast show.

Above ⤤
The PecanPi HAT features best-in-class components and dual DACs for superior audio reproduction

t's no surprise that we love microcontrollers here at HackSpace magazine. Their versatility and simplicity make them a must for electronics projects. Although a dab hand at reading sensors or illuminating LEDs, Arduinos and their friends do struggle when it comes to high-quality audio. If you need to add music or speech to your project, it may be worth getting a Raspberry Pi computer to do the heavy lifting. We're going to look at the various audio output options available for our favourite small computer, from a simple buzz, through to audiophile bliss.
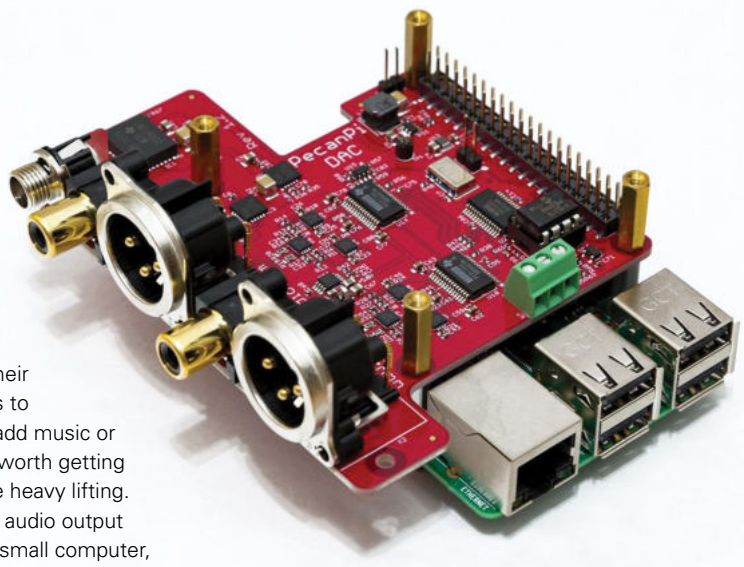
GET BUZZING

The simplest place to start is with the humble buzzer. A cheap active buzzer can be quickly added to Raspberry Pi's GPIO. It's surprisingly easy too. Try connecting a buzzer's red wire (positive) to GPIO pin 22 (Broadcom numbering) and the black wire (ground) to any GND pin. Now, install the GPIO Zero Python library by typing this at the command line:

```
sudo apt install python3-gpiozero
```

Create a file called **buzz.py** in your favourite editor and enter the following:

```
import time
from gpiozero import Buzzer
buzzer = Buzzer(22)
buzzer.on()
time.sleep(1)
buzzer.off()
```

Run it at the command line:

```
python3 buzz.py
```

YOU'LL NEED

◆ **Raspberry Pi 4**

◆ **Audio device(s) as featured**

You should hear a one-second buzz. See if you can make Morse code sounds by changing the duration of the `sleep` statement.

PASSIVE BUT NOT AGGRESSIVE

Raspberry Pi computers, with the exception of the Zero range, all have audio output on board. The original Raspberry Pi featured a stereo 3.5 mm socket, and all A and B models since feature a four-pole socket that also includes composite video. This provides your cheapest route to getting audio from your Raspberry Pi computer. A low-cost passive speaker can be directly plugged in to provide sound, albeit probably quieter than you'd like. Of course, add an amplifier or active speaker and you can have sound as loud as you like. This is the most direct way of adding sound to your project, but how to get the sound out?

Normally, the Raspbian operating system will recognise that an audio device has been connected and route audio through it. Sometimes, especially if you've connected an HDMI monitor with sound capability (e.g. an HDMI TV), sound will not come

out of the correct device. To fix this, open up a Terminal window and run `sudo raspi-config`. When the menu appears, go to 'Advanced Options' and select 'Audio', then select the option to force the output through the audio jack. You may need to reboot Raspbian for all changes to take effect.

PLUG AND PLAYBACK

A USB sound device is another simple choice for audio playback on Raspberry Pi. Literally hundreds are available, and a basic input/output device with better audio quality than the on-board system can be purchased for a few pounds online. Installation tends to be no more complicated than plugging the device into the USB port. You may need to select the new output, as the underlying audio system, ALSA (see

> **"**
> As they are low-power,
> USB devices **do not
> tend to feature
> amplification**
> **"**

box below), may mute it by default. To fix this, run `alsamixer` from the command line, press **F6** to select the new sound device, and if you see 'MM' at the bottom of the volume indicator, press **M** to unmute and adjust the volume with the cursor keys.

Unsurprisingly, when choosing your USB sound device, you can start at a few pounds and go right up to professional equipment costing hundreds. As they are low-power, USB devices do not tend to feature amplification, unless they have a separate power source.

LET'S PLAY

The simplest way to play audio on Raspbian is to use OMXPlayer. This is a dedicated hardware-accelerated

command-line tool that takes full advantage of Raspberry Pi's capabilities. It sends audio to the analogue audio jack by default, so playing back an MP3 file is as simple as:

```
omxplayer /path/to/audio/file.wav
```

There are many command-line options that allow you to control how the audio is played. Want the audio to loop forever? Just add `--loop` to the command. You'll notice that when it's running, OMXPlayer provides a user interface of sorts, allowing you to control playback from within the Terminal. If you'd just like it to run in the background without user input, run the command like this:

```
omxplayer --no-keys example.wav &
```

Here, `--no-keys` removes the interface, and the ampersand (`&`) tells the operating system to run the job 'in the background' so that it won't block anything else you want to do.

OMXPlayer is a great choice for Raspbian, but other players such as mpg321 are available, so find the tool that's best for you.

Another useful utility is speaker-test. This can produce white noise or vocal confirmation so you →

QUICK TIP

Need to adjust the volume from the command line? Just enter `alsamixer` to access a simple control screen.

Above
Small, cheap, and fun, the Speaker pHAT features a 3W speaker and LED VU meter

Below
Need a simple solution? USB audio devices come in all shapes and sizes but are mostly plug-and-play
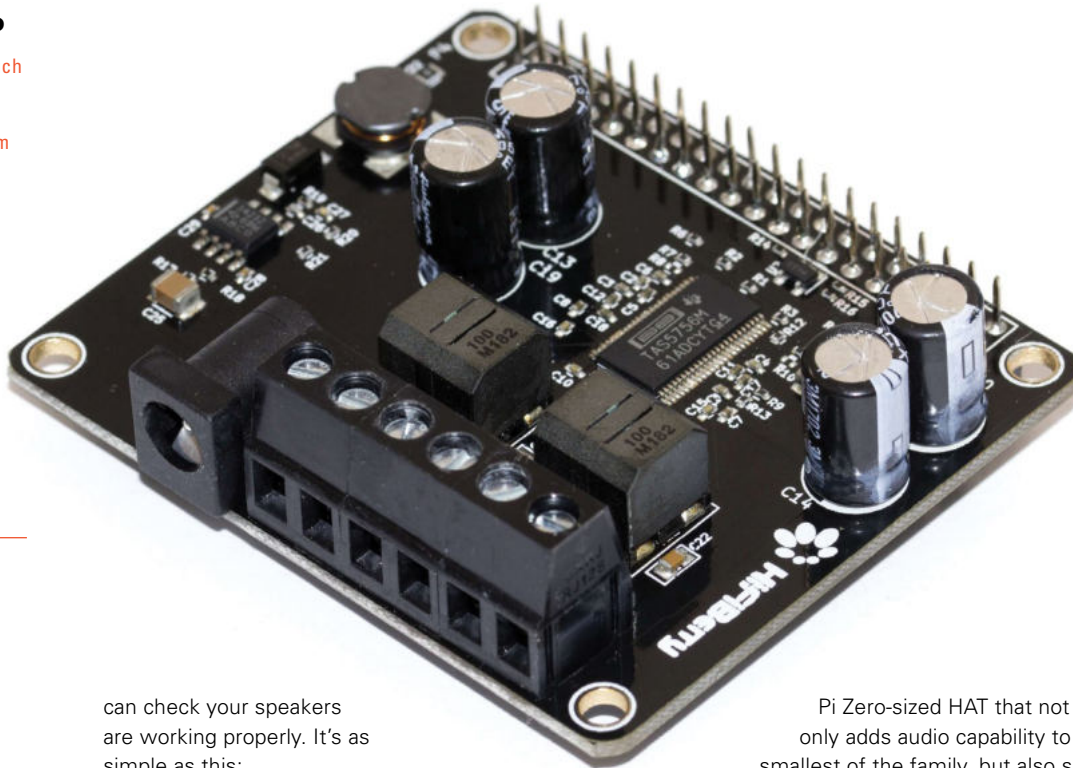
ALSA AND PULSEAUDIO

There's often confusion between these two systems. Raspbian comes pre-installed with ALSA (Advanced Linux Sound Architecture), which is the low-level software that makes sound work. It comes with a range of utilities to control output device, volume, and more. PulseAudio is a software layer that sits on top of ALSA to provide more features, including streaming capabilities. Chances are, if you need to do something a bit more clever than just play audio, you'll need to install a PulseAudio server.

Above
Many DACs also come with on-board amplifiers. Perfect for passive speakers

Below
Money no object? The Allo Katana is a monster DAC, and weighs in at £240, but outperforms £1000 equivalents

can check your speakers are working properly. It's as simple as this:

```
speaker-test -t wav -c 2
```

The first parameter sets the sound to be a voice, and the **-c** tests stereo channels only: front left and front right.

PHAT BEATS

If space is an issue, a Raspberry Pi 4, amplifier, and speaker may not be what you have in mind. After all, your cool wearable project is going to be problematic if you're trailing an amplifier on a cart with a 50-metre extension lead powering everything. Luckily, the clever people at Pimoroni have you covered. The Speaker pHAT is a Raspberry

FILE TYPES

There are command-line players available for just about every audio format in common use. Generally, MP3 provides the best balance of quality and space, but lower bit-rates result in lower sound quality. WAV is completely uncompressed, but can eat up your SSD card. If you don't want to compromise on audio quality, try FLAC, which is identical in quality to WAV, but much smaller. To convert between audio types, consider installing FFmpeg, a powerful audio and video processing tool.

Pi Zero-sized HAT that not only adds audio capability to the smallest of the family, but also sports a 3 W speaker. Now you can play any audio with a tiny device and a USB battery pack. The installation process is fully automated, so no messing around with drivers and config files. Once the script has completed, you can run any audio tool as before, and the sound will be routed through the speaker. No, the maximum volume won't be troubling any heavy metal concerts, but you can't knock the convenience and form factor.

PLAYING THE BLUES

An easy way to get superior audio quality using a Raspberry Pi computer is Bluetooth. Recent models such as Raspberry Pi 4, 3B+, 3B, and even the Zero W support Bluetooth devices, and can be paired with most Bluetooth speakers, including from the command line. Once connected, you have a range of options on size and output power, plus the advantage of wireless connectivity.

Setting up a Bluetooth connection, especially if you are using the command line, can be a little challenging (see the 'Bluetooth cheat sheet' box on the opposite page). There is a succinct guide here: **hsmag.cc/N6p2lB**. If you are using Raspbian Desktop, it's a lot easier. Simply click on the Bluetooth logo on the top-right, and follow the instructions to pair your device.
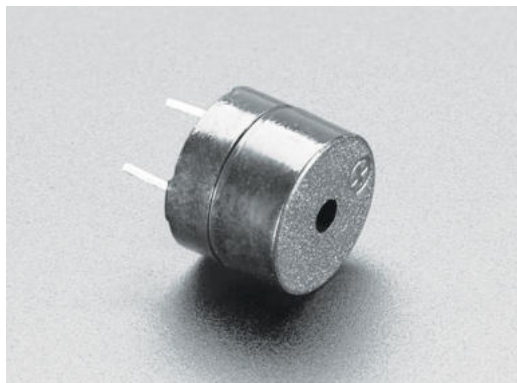
Above ◈
Need to keep it simple and under a pound?
A simple active buzzer is what you need

If you find OMXPlayer isn't outputting any audio, try installing mpg321:

```
sudo apt install mpg321
```

And try again:

```
mpg321 /path/to/audio/file.mp3
```

BUT SERIOUSLY

If your project needs good audio, and the standard 3.5 mm output just isn't cutting it, then it's time to look at the wide range of DACs (digital to analogue converters) available in HAT format. It's a crowded market, and the prices vary significantly depending on what you want from your device. Let's start at

> **It's a crowded market,
> and the prices vary
> significantly depending
> on what you want**

the lower end, with major player HiFiBerry's DAC+ Zero. This tiny HAT adds 192kHz/24-bit playback via two RCA phono ports for £12.50. If you're serious about your audio, then you may consider the firm's full HAT format high-resolution DAC+ Pro for £36, or really go for it with the DSP (digital sound processing) version for £67. All of these will require amplification, but the sound quality will rival audio components of a much higher price.

If money is no object and your project requires the best possible reproduction, then you can consider

BLUETOOTH CHEAT SHEET

If you want to pair a Bluetooth audio device (A2DP) on the command line, it can be a little hairy. Here's a quick guide:

First-time installation:

```
sudo apt-get install pulseaudio pulseaudio-module-bluetooth
sudo usermod -G bluetooth -a pi
sudo reboot
```

Start the PulseAudio server:

```
pulseaudio --start
```

Run the Bluetooth utility:

```
bluetoothctl
```

Put your speaker into pairing mode. Now, within the utility, run the following commands (pressing **ENTER** after each one):

```
power on
agent on
scan on
```

Now wait for the list to populate. When you see your device…

```
pair <dev>
```

Where **<dev>** is the displayed long identifier for your device. You can just type in the first few characters and press **TAB** to auto-complete. Do the same for the following steps.

```
trust <dev>
connect <dev>
```

Wait for the confirmation, then enter:

```
quit
```

Now try to play some audio using aplay (for WAV files) or mpg321 (for MP3). These instructions are adapted from the guide by Actuino at **hsmag.cc/N6p2IB**.

going full audiophile. There are some amazing high-end HATs out there, but one of the best-performing ones we've seen is the PecanPi DAC. Its creator, Leonid Ayzenshtat, sourced each individual component carefully, always choosing the best-in-class. He even used a separate DAC for each audio channel. The resulting board may make your wallet wince at around £200 for the bare board, but the resulting audio is good enough to be used in professional recording studios. If you've restored a gorgeous old radio back to showroom condition, you could do a lot worse than add the board in with a great amp and speaker. ▫

MAKER

Mike Cook

Veteran magazine author from the old days, writer of the Body Build series, plus co-author of *Raspberry Pi for Dummies, Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.

magpi.cc/mikecook

Part 03

Rotary encoders:
Create your own colour palettes

Use two rotary encoders to mix your own colours

You'll Need

> Two-knob rotary encoder box from *The MagPi* #93 **magpi.cc/93**

Last month we saw how to make a polar co-ordinate Etch-A-Sketch with a Raspberry Pi and two rotary encoders. Now, let's see how to use exactly the same hardware and explore the world of colour space.

01 Palettes

When we made the Polar Etch-A-Sketch last month, we used colour palettes from the Palettable library. This is good, but you did have to sort through their (admittedly large) array of palettes to choose the ones you wanted. We thought it would be good if you could design your own range of colours, and also learn a bit about colour theory as applied to computer graphics. Note that this is a rather large and specialist field, so we will be dipping our toe in the water at the simpler, more useful end of the topic, directly involved with drawing on a screen.

02 What are colours?

Colours are what the eye perceives in response to what wavelength of light is impacting the retina. When all the visible wavelengths of light (400 to 700 nm) are present, it sees white; when only a sub-band of this is viewed, it will see a colour. The longer wavelengths produce the sensation of red and the shorter ones of blue, with green being in the middle. With the colour receptors in the eye, the brain processes what colours you see, and this can be simulated by using a mixture of just three colour components.
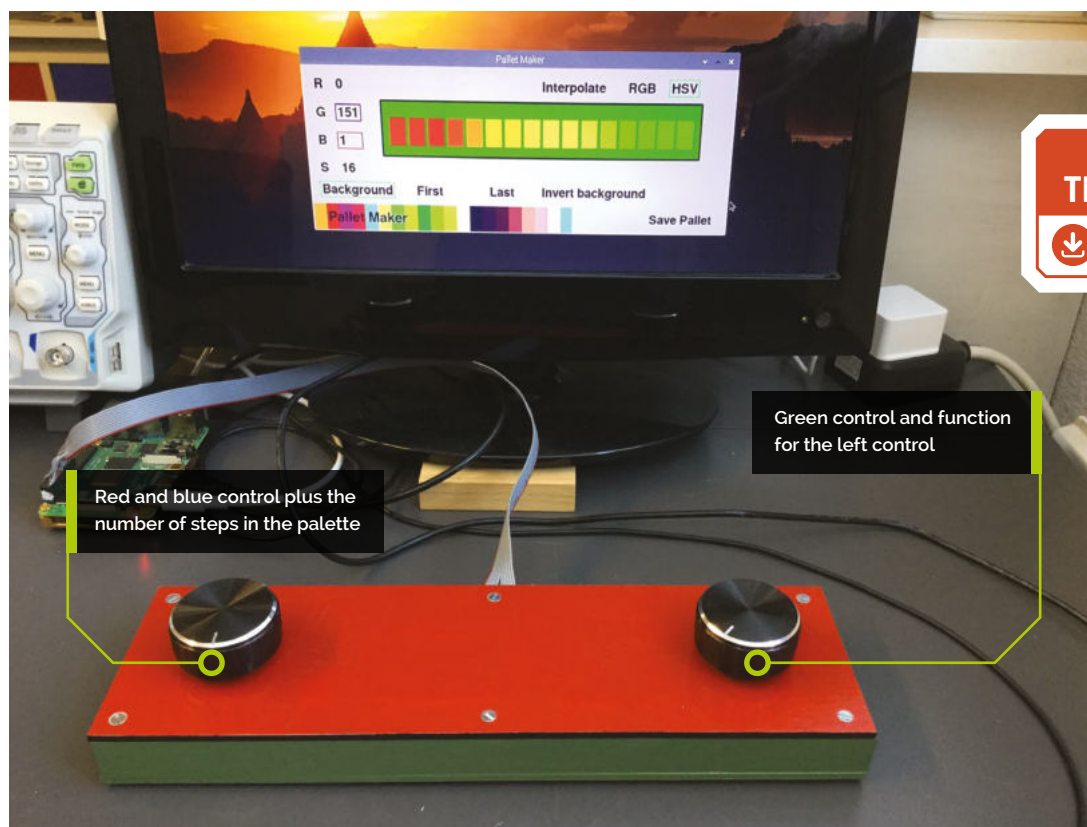
03 Mixing paint colours

There are two ways colours can mix, additively or subtractively, and which is applied depends on what you are mixing. For example, paint gets its colour from the colour of light it reflects. So yellow paint will reflect the band

Figure 1

▲ Figure 1 Blue and yellow paint mixed gives green
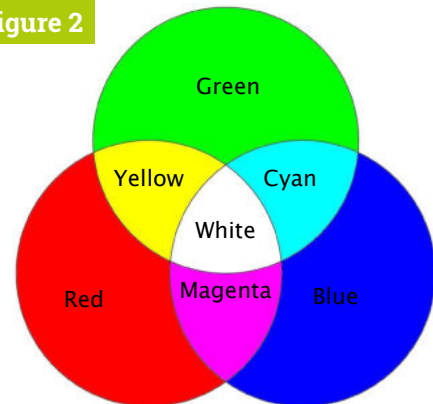
Figure 2

▲ Figure 2 RGB additive system using light

Red and blue control plus the number of steps in the palette

Green control and function for the left control

of wavelengths making up yellow (about 500 to 600 nm) – it subtracts all other wavelengths from white light shining on it. Blue paint covers a band approximately from 425 to 530 nm. So when you mix yellow and blue paint together, it only reflects the wavelengths that overlap these two bands, which we see as green (**Figure 1**).

> " The red, green, and blue lights are said to be additive primary colours "

04 Mixing light

Light, on the other hand, is additive. That is, if you shine three lights – a red, green, and blue one – together onto one spot, the eye will see this as white light. This is summarised in **Figure 2**. The red, green, and blue lights are said to be primary colours – or, more accurately, additive primary colours. Where just two primary colours overlap, you get a secondary colour: yellow, magenta, or cyan. These are the additive secondary colours. Note that this is the exact opposite of subtractive colours where the primary colours are yellow, magenta, and cyan, with the secondary colours being red, green, and blue.

Figure 3
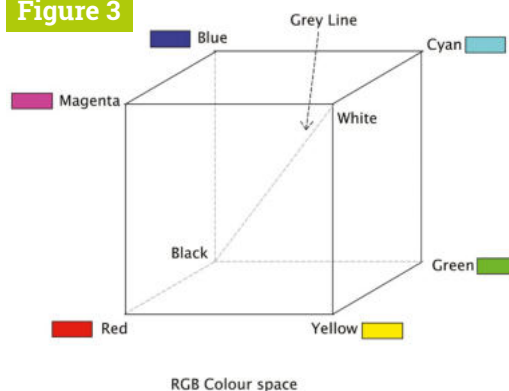
▲ **Figure 3** RGB additive system using light

05 Putting numbers on colours

The standard way of expressing a colour on a computer is by using a brightness factor for each of the three additive primary colours: red, green, and blue. The can be standardised to be a number between 0 – no component, to 1 – full component. But, more commonly, these values are represented by 0 to 255, because that range of numbers will fit into a byte. So as each colour is represented by three numbers, we can think of them as co-ordinates to a point in three-dimensional space. This gives us 'cubic colour space', as shown in **Figure 3**.

Top Tip 👍
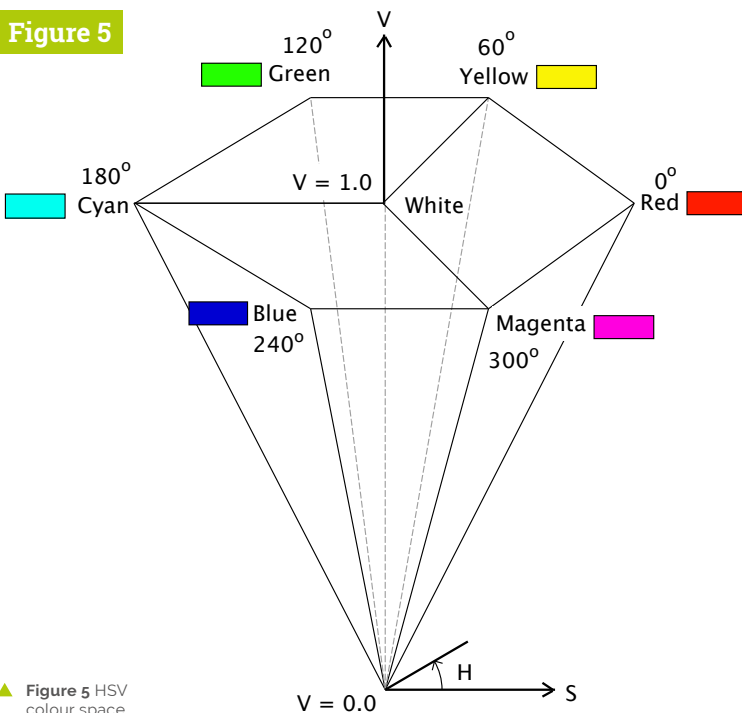
Understanding HSV space

How far round the cone, an angle, gives you the hue or colour. How far away from the centre gives you the saturation, or depth of colour. And how far down the cone gives you the value, or brightness, of the colour.
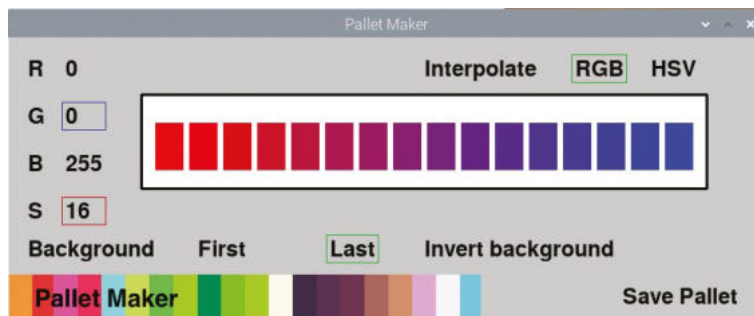
Figure 4
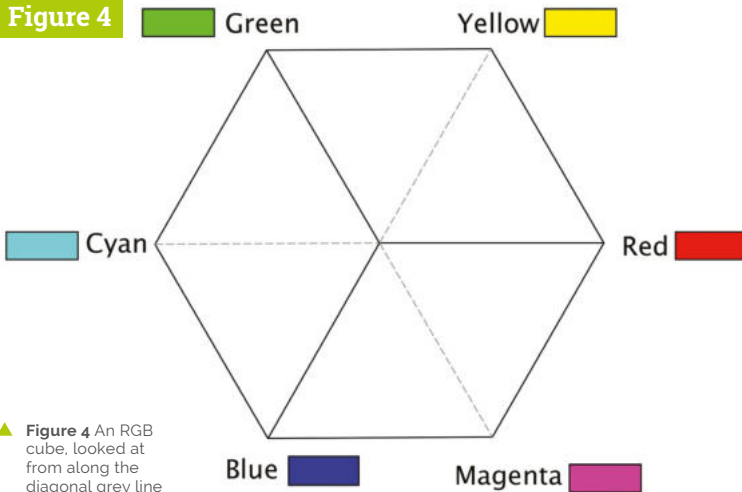
▲ **Figure 4** An RGB cube, looked at from along the diagonal grey line

Figure 5

▲ **Figure 5** HSV colour space

▲ Red to blue in RGB colour space

06 Colour cube

Each vertex of the cube represents a pure primary or secondary colour. Black is the vertex at the back, the point where red, green, and blue are all zero. This is the origin of the 3D space. The vertex diagonally opposite black is the point where all primary colours are full on; this represents white. A line joining the black and white vertex is known as the grey line. Any colour that sits on this line is a shade of grey; the closer it is to the white point, the lighter grey it is. Any point inside this cube represents a colour.

07 A path in cubic space

If we take any two points in cubic colour space and join them with a line, points on that line represent intermediate colours produced when fading between the two colours represented by the original two points. If we take a number of evenly spaced points along that line, the colours produced are a palette of shades between these two colours. Given that in an 8-bit colour cube you have 256 × 256 × 256 = 16,777,216 points or colours, then the number of lines you can possibly have in this space is quite staggering.

> **"** The path between two points depends on the geometric nature of the colour space **"**

08 A different space

Now if you look at the RGB cube along the axis of the grey line, you will see a hexagon (**Figure 4**). It is still a cube: you can tell if you look at the dotted hidden lines, and the solid visible lines joining the vertices of the cube. But now if you project that view onto a plane, and add a third dimension to represent the intensity, you get the HSV hexagonal cone model (**Figure 5**). Any point in cubic colour space can be transformed into HSV space with some simple maths.

09 So why bother with HSV?

The HSV colour space contains all the colours of cubic space, but in a different

arrangement. So when you draw a path between two colours in HSV space, the path – and hence the palette of colours you get – is different from that you would get if you connected the colours in cubic space. In other words, the path between two points depends on the geometric nature of the colour space. So, our software to generate a palette gives you the choice of either of these colour spaces to generate your palette in. It can be downloaded from our GitHub page (**magpi.cc/pibakery**).

10 Controls

Our software uses both a mouse and the two rotary encoders to control its actions. Basically, you only work on setting the first and last colours, then all the intermediate colours are generated automatically. The background colour is also controllable, and there is an option to invert this. We made the controls as intuitive as possible to use (**Figure 6**). Note that the push capability of these rotary controllers is used to change what aspect the rotary part is controlling. All the colour controls wrap round; that is, one negative click from zero brings a value of 255.

11 Using the software

You can set the parameter S, which is the number of steps or colours there are in the palette. You can have up to 16 steps. When you have made a palette and seen what it looks like on various backgrounds, you can save it to a file. We have used the .pal file format for this, as is used in some other applications, like GIMP. The file has a very simple structure, as shown in **Figure 7**: it consists of a header, version number, how many colours there are, followed by the RGB triplets.

Using the palettes

In order to help you use these palettes in your own code, we have written a small program that simply reads a .pal file and displays it – again, it is on our GitHub page. There are lots of other colour spaces for various specific applications, such as the CMY or CMYK (cyan, magenta, yellow, black) subtractive system used for colour printing. Or YUV, which is one luminosity or grey value Y, and two colour components U and V, used for generating colour images on old analogue TVs. **M**
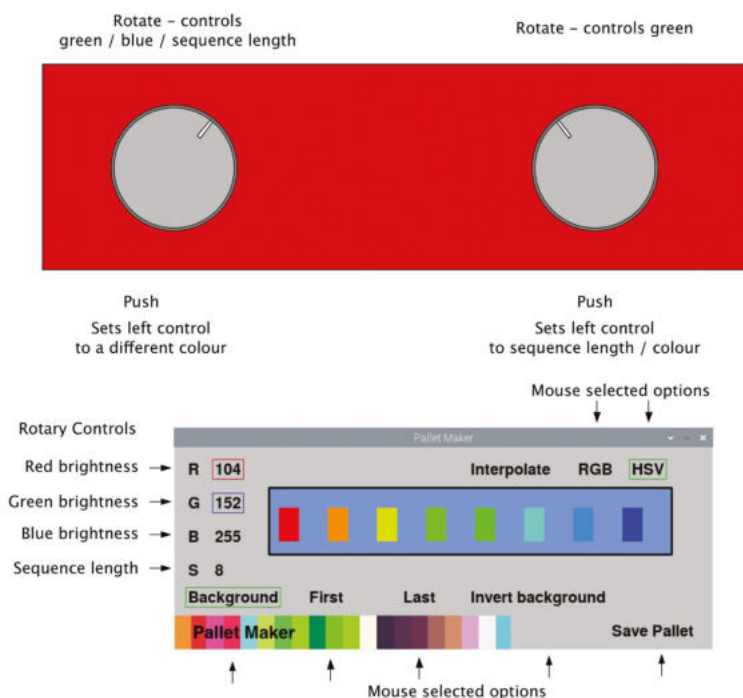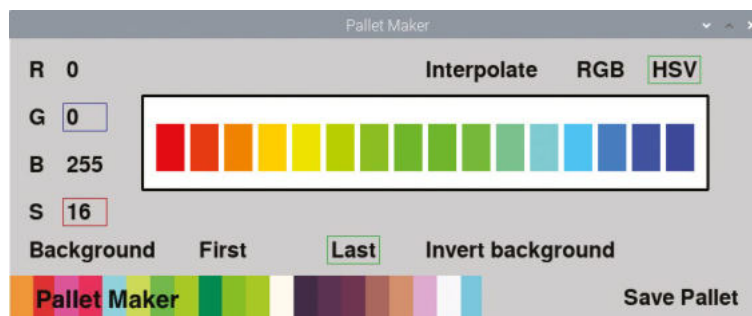
▲ **Figure 6** Physical controls and software user interface
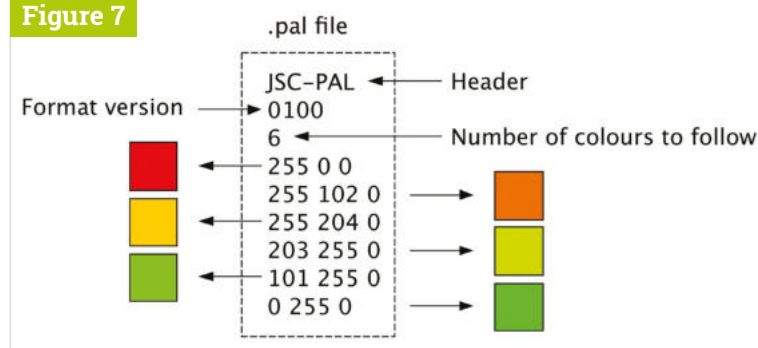
▲ **Figure 7** Format of the .pal palette file

▲ Red to blue in HSV colour space

High Quality Camera Projects

What to do with your brand new Raspberry Pi camera accessory

E ver since the High Quality Camera came out a month ago (on the same day as *The MagPi* #93), we've seen a lot of people get quite excited about the possibilities of using the new camera with a Raspberry Pi. We've also seen a lot of people turn their excitement into a project already.

This inspired us to look at some past Raspberry Pi Camera Module projects, to see just what could be upgraded and the benefits they might bring. No 3D-printing a lens adapter required!

User photo
gallery

Here are some of the amazing photos people have already taken with the HQ Camera

Seeing detailed shots of the moon always make us excited. This one was taken by Mark S Hurst (**@Scoundrel666**)

Raspberry Pi HQ Camera - Venus

0mm lens @ 600mm   + 2x Teleconverter   Scaled x4
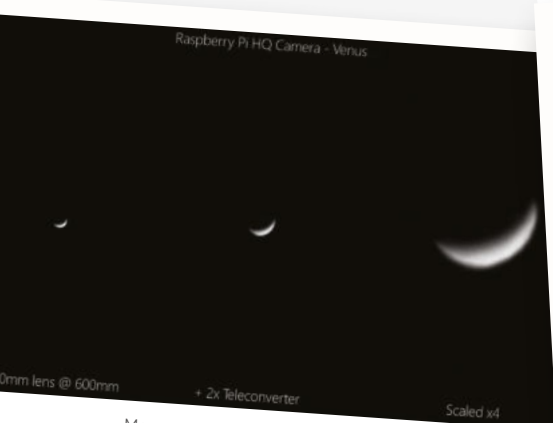
More astrophotography – this video of a Venus flyby taken by Movie Vertigo (**@MovieVertigo**) is absolutely astonishing. See more here: **magpi.cc/vertigo**

Beautiful sunset shot using the 16 mm lens by Dave Hunt (**@climberhunt**)
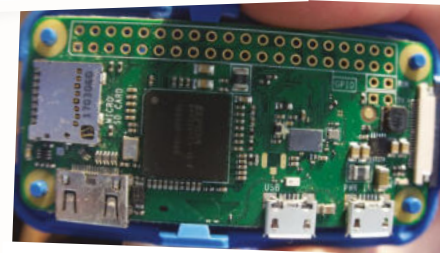
A wonderful macro shot of a Raspberry Pi Zero by Peter Onion (**@PeterOnion**)

Richard Hayler (**@rdhayler**) took this as a video using his special camera setup – see the tutorial in this feature

Thank you to Martin Mander (**@OldTechNewSpec**) for taking this video of a magpie using a Raspberry Pi so we could feature it in *The MagPi*

We think Raspberry Pi Digital Content Manager Alex Bate (**@alexjrassic**) really likes The Child

Projects to upgrade

Power up these amazing camera projects with an HQ Camera

Formula Pi/ computer vision

The Formula Pi racing series makes use of computer vision so that robot cars can automatically race around a specially prepared track. It's not the only style of project to use computer vision, machine learning, or AI in this way, all of which will benefit from the improved quality of the HQ Camera.

formulapi.com

Naturebytes

An outdoor wildlife camera that makes use of a Raspberry Pi camera, it has a PIR sensor that it uses to track when motion occurs. We've seen some great shots of birds and beasties throughout the years, and we can't wait to see even higher-quality photos.

naturebytes.org

Ping video doorbell

A Martin Mander classic upcycling of a 1986 intercom to act as a video doorbell that calls the home-owner using Google Duo when the button is pressed. You can talk to the person at the door while seeing who they actually are. It will work a lot better with a higher-quality camera.

magpi.cc/ping

Parent detector

This great project from the Raspberry Pi projects site is a good way for younger makers to take their first steps in camera coding, and upgrading it with an HQ Camera means they can adapt it into many other uses.

magpi.cc/detector

Webcam

Last issue we showed you how to use a Raspberry Pi Camera Module as a webcam while working from a Raspberry Pi – upgrading it with an HQ Camera is a bit of a no-brainer, especially with HD webcams scarce on online stores.

magpi.cc/93

High Quality
Camera case hack

Turn an official Raspberry Pi case into the ultimate hackable camera rig

MAKER

Ozzy and Richard

Ozzy Hayler-Goodall was one of the winners of the first Astro Pi competition back in 2015. He likes to hack things and run 5K Parkruns with his dad, Richard. They also like to eat chocolate.

@rdhayler

With the release of the new High Quality Camera, the opportunities for taking high-quality pictures with a Raspberry Pi computer have grown dramatically. But you don't always want to have to take a keyboard and mouse along with you. Using a mini LCD screen and a hacked official Raspberry Pi case, we're going to show how to make a hand-held camera capable of taking great pictures.

You'll Need

> Official Raspberry Pi 3 case
magpi.cc/pi3case

> Adafruit PiTFT touchscreen 2.8" or 2.4" (or any other screen) with buttons
magpi.cc/pitftplus

> High Quality Camera, lens, and ribbon cable
magpi.cc/hqcamera

> Push-button

> Jumper wires

> Tools (drill, rotary tool, glue)

> 4 × M2.5 8 mm screws and nuts

> Five-pin female to male header (if using bigger screen)

01 Installing the screen
The first step is to connect the screen to the Raspberry Pi. To start, you need to mount the screen onto the GPIO pins. Once it's connected, make sure you have a microSD card with the latest version of Raspbian. Boot up Raspberry Pi and open a Terminal window. There, if using an Adafruit screen, you need to type:

```
wget https://raw.githubusercontent.com/
adafruit/Raspberry-Pi-Installer-Scripts/
master/adafruit-pitft.sh

sudo bash adafruit-pitft.sh
```

Figure 1

▲ **Figure 1** Make sure you clamp the case, or carefully use a vice when drilling. Print out the template and stick it to the case

Then you need to answer the questions based on which screen you have and how you want to use it. The Adafruit website has a comprehensive tutorial with more detailed instructions for installing the screen: **magpi.cc/pitftguide**.

Once the installer script has finished, reboot and you should see the Raspbian desktop appear on the small screen.

02 Hacking the official case – base
Wear gloves and goggles when modifying the case!
The case shown here is for a Raspberry Pi 3, but the steps will be the same for a Raspberry Pi 4 version. First, mark out the holes on the bottom of the case. If you do not want to measure it out yourself, there's a template that you can print out in the GitHub repo for this project (**magpi.cc/hqchack**). Drill the four corner holes with a 2.5 mm bit, then the two inner holes with a 4.5 mm bit, as shown in **Figure 1**. Finally, cut out a slot for the camera cable connector with your rotary tool or a craft knife.

> ❝ Make sure you have a microSD card with the latest version of Raspbian ❞

03 Hacking the official case – top
Use your rotary tool to carve out a rectangle for the screen from the red top part of the case. It should be big enough to see the screen and allow access to the buttons (if your screen has them). If you're using the 2.8-inch screen, then you'll need to remove most of the top face of the case.

The last case modification is to drill two holes for the button wires in the white side panel with the port openings.

A push-button used to trigger the camera, like on a traditional camera

The High Quality Camera with a 16 mm lens attached

Modified official Raspberry Pi case. Use caution and adult supervision when modifying
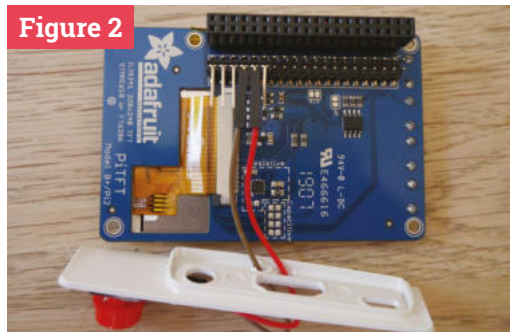
04 Connecting the button

The external button is used as the main control for taking a picture. You need to connect two wires: one to a GPIO pin and one to a ground pin. The Python code provided uses the GPIO 12 pin. If you are using the bigger 2.8-inch screen, you can use a five-pin 90-degree header (**Figure 2**) to attach the wires to the pins, otherwise they won't fit. Thread the wires through the two holes on the side of the case and then, with a blob of hot glue, attach the button to the white outside panel of the case.

Figure 2

▲ **Figure 2** There isn't enough room to attach jumper wires to the pins on the screen. If you don't want to solder on wires, use a 90-degree header

05 Connecting the camera

Connect the ribbon cable to the connector on the HQ Camera board. Then pass it through the slot that you cut in Step 02. Now use the M2.5 screws to attach the HQ Camera board to the outside of the bottom piece of the case.

Fold over the cable at 90 degrees, as shown in **Figure 3** (overleaf). Then insert the Raspberry Pi into the bottom of the case, and connect the other end of the cable. You should find that a standard 16 cm camera cable fits perfectly. Make sure you follow the usual steps for connecting the cable, and that you are putting it in the right way around.

06 Software

Open a Terminal and clone the project using the command:

```
git clone https://github.com/crocodile54/
HQC-case-hack.git
```

This will create a folder which contains the Python program and the desktop backgrounds for the user interface.

The first part of the code **hqc_hack.py** is importing the libraries. The GPIO Zero library is used to control the buttons. At the time of

Warning!
Power tools

Please use safety goggles and gloves when drilling, sawing, and sanding

hqc_hack.py

> Language: **Python 3**

```python
001.    from gpiozero import Button
002.    from picamera import PiCamera
003.    from signal import pause
004.    import subprocess
005.    import time
006.
007.    BACKDIR = "/home/pi/HQC-case-hack/images"
008.    DATADIR = "/home/pi/Pictures"
009.    # Adafruit TFT buttons
010.    b_bot1 = Button(27)
011.    b_top4 = Button(17)
012.    b_2 = Button(22)
013.    b_3 = Button(23)
014.    # External button on GPIO 12
015.    b_ext = Button(12)
016.
017.    # operation modea
018.    video = False
019.    burst = False
020.
021.    camera = PiCamera()
022.    # Limit preview window size to reveal button labels on screen
        background
023.    camera.start_preview(fullscreen=False, window=(0, 10, 580,
        480))
024.    prev_active = True
025.
026.    def set_backgrounds(back):
027.        """Calls the pcmanfm command to change the desktop
        wallpaper."""
028.        bfile = BACKDIR + "/panel_" + back + ".jpg"
029.        subprocess.call("pcmanfm --display :0 --set-wallpaper " +
        bfile, shell = True)
030.
031.    set_backgrounds("inactive")
032.
033.    def video_mode():
034.        """Run when a button is pressed: sets the mode to
        video"""
035.        global video
036.        global burst
037.        if video:
038.            video = False
039.            burst = False
040.            set_backgrounds("inactive")
041.        else:
042.            video = True
043.            burst = False
044.            set_backgrounds("video")
045.
046.    def byebye():
047.        """Run when a button is pressed: Shuts down the Raspberry
        Pi"""
048.        set_backgrounds("halt")
049.        subprocess.call("sudo nohup shutdown -h now", shell=True)
050.
051.    def blank():
052.        """Run when a button is pressed: Blanks/unblanks the
        screen"""
053.        global prev_active
054.        global camera
055.        if prev_active:
056.            camera.stop_preview()
057.            prev_active = False
058.            subprocess.call("xset -display :0 dpms force off",
        shell=True)
059.        else:
060.            camera.start_preview(fullscreen=False, window=(0, 10,
        580, 480))
061.            subprocess.call("xset -display :0 dpms force on",
        shell=True)
062.            prev_active = True
063.
064.    def burst_mode():
065.        """Run when a button is pressed: sets the mode to
        burst"""
066.        global burst
067.        global video
068.        if burst:
069.            burst = False
070.            video = False
071.            set_backgrounds("inactive")
072.        else:
073.            burst = True
074.            video = False
075.            set_backgrounds("burst")
076.
077.    def take_photo():
078.        """Run when a button is pressed: Captures data based on
        operation mode"""
079.        global camera
080.        global prev_active
081.        blank()
082.        camera.close()  # Closes PiCamera connection
083.        if video:  # take a 10 second video
084.            filename = DATADIR + "/vid" + str(
        time.strftime("%Y%m%d-%H%M%S"))+".h264"
085.            subprocess.call(["raspivid",
086.                            "-t", "10000",
087.                            "-b", "15000000",
088.                            "-fps", "30",
089.                            "-vs",
090.                            "-p", "0,10,580,480",
091.                            "-o", filename])
092.        else:
093.            if burst:  # Take 10 rapid still images
094.                subprocess.call(["raspistill",
095.                                "-t", "8500",
096.                                "-tl", "850",
097.                                "-o", DATADIR + "/img%02d.jpg",
098.                                "-p", "0,10,580,480",
099.                                "-bm",
100.                                "-dt"])
101.
102.            else:  # Take a single still image
103.                filename = DATADIR + "/img" + str(
        time.strftime("%Y%m%d-%H%M%S"))+".jpg"
104.                subprocess.call(["raspistill",
105.                                "-o", filename,
106.                                "-p", "0,10,580,480",
107.                                "-ex", "antishake"])
108.        camera = PiCamera()  # re-connect to Camera using
        PiCamera
109.        blank()
110.
111.    # Set functions to be run when buttons pressed
112.    b_ext.when_pressed = take_photo
113.    b_top4.when_pressed = blank
114.    b_2.when_pressed = video_mode
115.    b_3.when_pressed = burst_mode
116.    b_bot1.when_pressed = byebye
117.    pause()  # Keep the program active
```

▲ **Figure 3** When folding the camera cable, make sure it isn't blocking the microSD card slot

writing, the picamera library does not support the maximum resolution of the new camera, so it is only used for previewing the shot. Instead, the `raspistill` command is used to capture images and movies, and is run using the subprocess library – which is also used for changing the background and blanking the screen.

To stop the image files being overwritten every time you take a new picture, the time library is used to create a unique file name that also shows when the photo was taken. The signal library is also used to keep the program running.

07 Customising the buttons

If you are using the 2.4" or 2.8" Adafruit screens, you will have noticed that they come with buttons built in alongside the screen. These buttons can be used for many different things and you can customise them by changing the code. In **hqc_hack.py** they are used for switching to video mode, switching to burst mode, blanking the screen (to save battery life), and shutting down Raspberry Pi. If you plan to use your camera mainly for taking videos, then your buttons could choose the length of the video (in **hqc_hack.py**, video capture only lasts for ten seconds).

Top Tip 👍

Start on boot

To make your code run when the Raspberry Pi starts, add the following line to the **/etc/rc.local** file, before the `exit` line:

```
sudo -p pi python3 /home/pi/HQC-case-
hack/hqc_hack.py &
```

08 Simple user interface

As well as a live view, the screen can show what each button does and which mode is selected. You can do this by changing the desktop wallpaper to have words next to each button explaining what it does. However, if you are planning to change the backdrop, you will also have to change the size of the preview window so you can still see the words next to the buttons. The GitHub repo contains an **images** folder which has the backgrounds used in **hqc_hack.py**, as shown in **Figure 4**.

If you have problems with the program, there is a debug version also in the repo.

> ❝ If you are planning to change the backdrop, you will also have to change the size of the preview window ❞

09 Battery pack

Using a battery back makes the camera portable. You can use a phone battery pack, as a LiPo cell won't be able to power it for long. And even if you are using a phone charger, it is a good idea to take a spare in case it runs out. You can just hold the battery or attach it with some sticky Velcro to make it more secure. 🅜
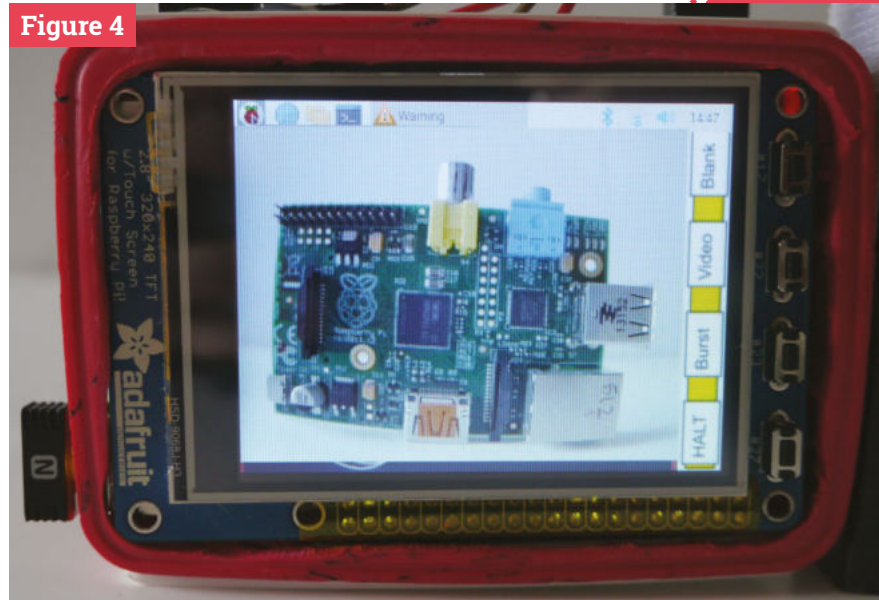
▲ **Figure 4** The desktop wallpaper on the screen is used to show the function of each button, but you can customise yours for different tasks

Strato Pi **CM Duo**

Keeps your system running during updates with not one, but two microSD cards. By **Lucy Hattersley**

SPECS

CASE:
Two-module DIN rail cass

POWER:
9–28 V DC power supply, with surge and reverse polarity protection

INPUT/ OUTPUT:
10/100 Ethernet port; two USB 2.0 type-A ports, with individual power control and fault detection; standard RS-485 interface

Strato Pi CM Duo is an industrial server comprising a Raspberry Pi Compute Module 3+ inside a DIN-rail case with a slew of additional features. Not least of which is the presence of two microSD card slots (hence the 'Duo' moniker).

As a quick reminder, a Compute Module is the innards of a Raspberry Pi 3 computer integrated onto a small (67.6 × 31 mm) board that plugs into a standard DDR2 SODIMM connector (**magpi.cc/cm3plus**).

Strato Pi takes this stripped-down Raspberry Pi and uses the DDR2 SODIMM connector to hook it up to a custom board with 10/100 Ethernet, two USB Type-A ports (with individual power and fault detection), a real-time clock and CR1025 battery, and a hardware watchdog chip (used to switch

▶ The green plastic terminal block is used to wire power to the device and implement an RS-485 port

Verdict

Strato Pi CM Duo is an excellent piece of kit. We found it to be well-built, with an intelligent design that integrates Raspberry Pi with the industrial landscape. It's not a low-cost solution, but then neither is industrial downtime.

8/10

❝ The primary use-case is to maintain uptime during any upgrade process ❞

between the two microSD cards and perform a hardware reset if required).

A green plastic terminal block has five positions for power and serial connections. It's rated 9–28 volts with surge protection (and delivers 1.9 A at 5 V to the Compute Module).

The Strato Pi takes over most of the GPIO pins, but uses the standard UART TX/RX pins on the GPIO connector to implement a standard RS-485 port. The RS-485 interface can be used to integrate the Strato Pi with a range of industrial control systems and communication signals. It

can handle up to 32 devices, at a range of up to 1200 metres.

Uptime

Our device shipped with stock Raspbian, and we removed one of the microSD cards and added a blank **ssh** file to the boot system to gain access (don't forget to change the password). From there you can install the Strato Pi utility software, install the real-time clock software, and control the RS-485 serial port. Detailed instructions are in the user guide (**magpi.cc/cmduoguide**).

The two microSD card slots are hidden inside the case (away from prying fingers) and you can switch between the two. You can run one as a boot drive, and the second as storage; but the primary use-case is to maintain uptime during any upgrade process. **M**

CUSTOM PC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING

THE MAGAZINE FOR

PC HARDWARE ENTHUSIASTS

ISSUE 202 OUT NOW

VISIT **CUSTOMPC.CO.UK** TO LEARN MORE

HackSpace

TECHNOLOGY IN YOUR HANDS

THE MAGAZINE FOR THE MODERN MAKER

ISSUE #31

OUT NOW

hsmag.cc

Unicorn **HAT Mini**

▶ Pimoroni   ▶ **magpi.cc/unicornhatmini**   ▶ £20 / $20

Light up your world with this mini multicoloured marvel. By **Phil King**

Building on the success of its Scroll HAT Mini board, which features a matrix of bright white LEDs, Pimoroni has launched the Unicorn HAT Mini: a multicoloured version with individually addressable RGB pixels.

It also carries the Unicorn branding, as seen previously on the original 8×8 Unicorn HAT, 16×16 Unicorn HAT HD, and supersize Ubercorn. This time the Zero-size board features a 17×7 matrix.

❝ The frame rate test averaged an impressive 67fps ❞

One slight downside is that – unlike the full-size Unicorn HAT HD – it doesn't come supplied with a translucent diffuser layer to fill out the gaps in the display between the small pixels. However, you could always make your own, even out of paper.

Push-button controls

Four programmable push-buttons – two on either side of the matrix – feel satisfyingly springy and responsive, while their input can be read using the standard GPIO Zero library.

Since the Unicorn HAT Mini uses different pixels and twin LED driver chips

▲ A dazzling display from the matrix of individually addressable RGB LEDs

from the other Unicorn boards, examples and code written for those won't run directly without a few tweaks to the code first.

Fortunately, Pimoroni has put together a special Python library for the Unicorn HAT Mini (**magpi.cc/unicornhatminigit**). Note that you'll need to enable SPI manually (in Raspberry Pi Configuration > Interfaces) to get it to work.

As well as enabling precise individual pixel control using RGB and brightness values, the library features eleven Python code examples to get you started. These include a nifty pixel shading demo, multicoloured scrolling text, and fun Simon and Columns to make use of the buttons. To give you an idea of how fast the matrix can be refreshed, an 'fps' example tries to do so as rapidly as possible while printing the frame rate to the Terminal – in our tests, it averaged an impressive 67fps.

Of course, you can always write your own code to do whatever you like with this versatile little board. Robin Newman has even managed to control its display using OSC messages from Sonic Pi: **magpi.cc/unicornsonicpi**. ▪

◀ The underside of the board reveals the two LED driver chips which ensure a fast refresh rate
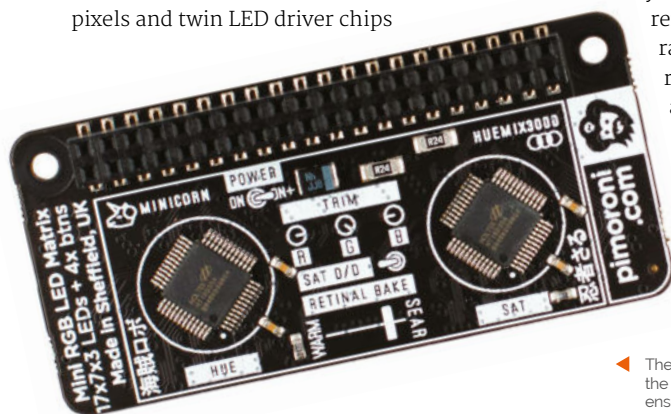
SPECS

DISPLAY:
17×7 RGB matrix (119 LEDs)

LED DRIVER CHIPS:
2 × Holtek HT16D35A

INPUTS:
4 × tactile buttons

DIMENSIONS:
65×30×10 mm

Verdict

A versatile mini RGB LED matrix with bright pixels, a fast refresh rate suitable for animations, and the bonus of four programmable push-buttons.

9/10

10 Best:

Raspberry Pi robot kits

Create an automaton with these excellent kits for Raspberry Pi

We've said it before, but the way Raspberry Pi has helped create a revolution in hobby electronics has always excited us. We've made a few robots here in the pages of the magazine as well; however, you don't need to do it from scratch. With a wealth of robot kits to get you started, here are ten of our favourites. **M**

▲ PiCar-S   £98 / $120

Fully specced robo-car

This sensor-packed 'smart car' looks very DIY and we dig it. The horns on the ultrasonic sensor are a nice touch as well.

magpi.cc/picars
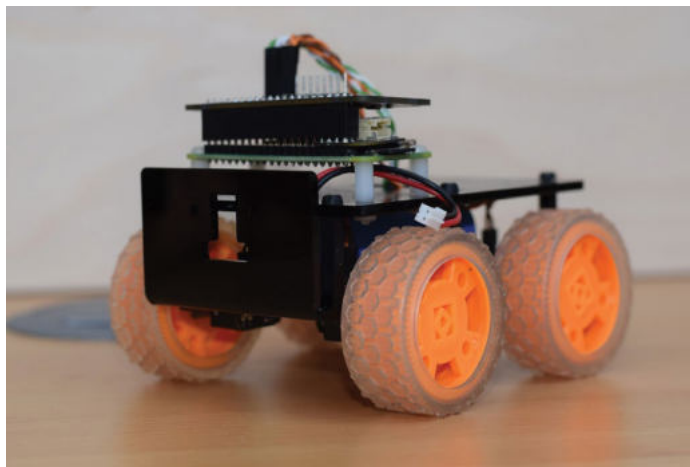
▲ Tiny 4WD   £55 / $67

MagPi-inspired robotics

The design for the Tiny 4WD actually originated in *The MagPi* a few years ago. The Tiny 4WD you can buy is a souped-up version, and you won't need to laser-cut your own parts.

magpi.cc/tiny4wd

▸ GoPiGo 3 £95 / $116

Robotics for classrooms

The GoPiGo is a fairly basic robot platform that can be expanded upon pretty well, and also comes in special packs for classrooms so many people can build and learn at once.

magpi.cc/gopigo3
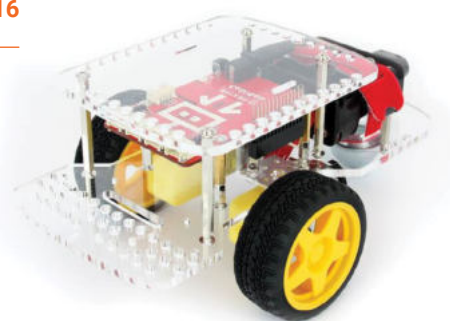
◂ Picoh   £100 / $123

Cute roboface

We quite liked the big Ohbot face when it came out, so we were excited when the digital Picoh arrived with a far cuter face. It's simpler to assemble and can easily be controlled by a Raspberry Pi.

magpi.cc/picoh
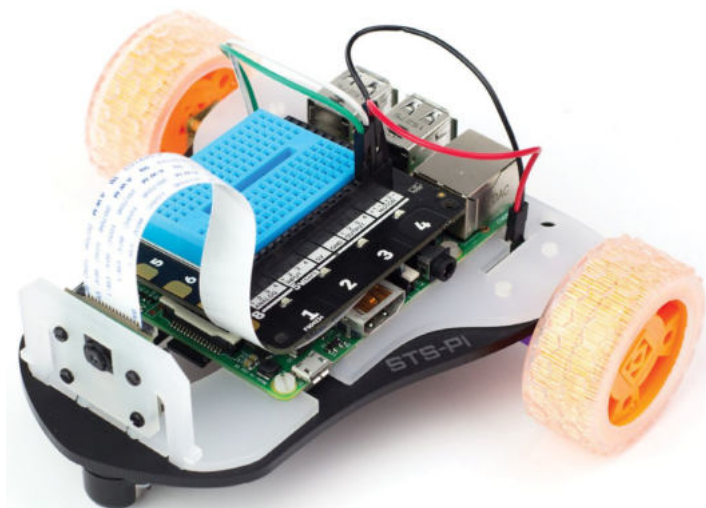
▲ STS-Pi

£28 / $29

Cheap and cool

While slightly more expensive than the CamJam EduKit #3, it does include a chassis you can build upon. Plus, it looks a bit like a Space Shuttle.

magpi.cc/stspi

▶ AlphaBot2 £95 / $117

A towering robot

While not actually towering, it is a two-tiered round robot with a very fancy setup. You could easily upgrade it with a High Quality Camera as well for remote photography.

magpi.cc/alphabot2
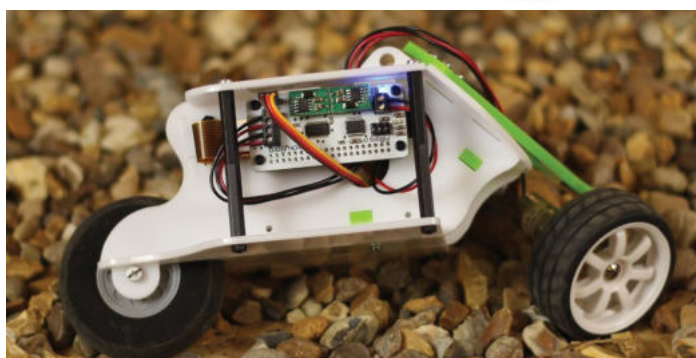
▲ RockyBorg

£99 / $135

Fun on three wheels

The cheaper PiBorg robot option, the RockyBorg is an incredible and very nippy little three-wheeled robot. It banks like a bike when it turns, which immediately makes it 50% cooler.

magpi.cc/rockyborg

▲ CamJam EduKit #3

£18 / $22

Robotics for beginners

Still the best beginner kit for folks wanting to make a robot with a Raspberry Pi, the CamJam EduKit #3 makes it relatively easy and gives you loads of transferable skills for any future robotics project.

magpi.cc/edukit3

▲ MonsterBorg

£210 / $286

Self-proclaimed ultimate robot

This incredible robot kit has been used in the excellent Formula Pi AI racing league, where people submit code that is then uploaded to one of these incredibly sturdy and robust wheeled robots.

magpi.cc/monsterborg

▶ MeArm Pocket

£33 / $40

Small yet mighty

This newer, smaller version of the classic MeArm robot arm kit can be controlled with a Raspberry Pi or microcontroller. We still think it's very cool.

magpi.cc/pocketarm
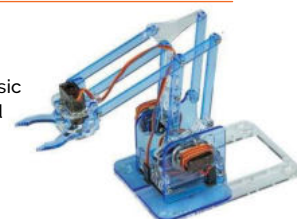
Learn Python
with Raspberry Pi

Get started with Python programming using a Raspberry Pi and these resources. By **Lucy Hattersley**

Code Club: Python Module 1

Python is an incredibly powerful and popular programming language. You can use it to control Raspberry Pi and build programs that interact with a range of hardware and software.

Despite its versatility, the programming language is very easy to pick up and learn, which is why almost all of the projects you find in *The MagPi* use the Python language.

The Raspberry Pi Foundation is deeply involved with teaching Python to a new generation of coders (of all ages), and to that end you'll find a huge array of projects on its website. Code Club is part of Raspberry Pi and a lot of its focus has gone online, including this cracking Python: Module 1 course.

Here you'll find a range of projects, from creating your own 'Rock, Paper, Scissors' game to making colourful posters with code. It's a fun course and the perfect introduction to coding with Raspberry Pi. **M**
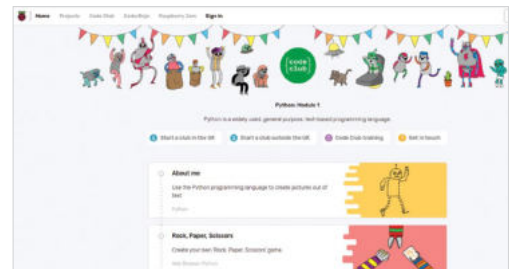
Documentation

Bookmark these sites when learning Python
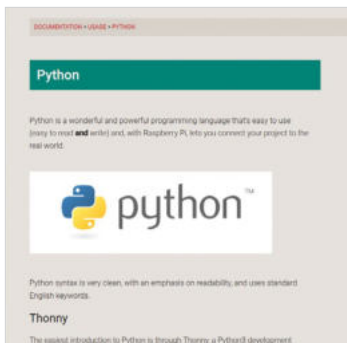
RASPBERRY PI PYTHON DOCUMENTATION
Make sure you take a look at the Raspberry Pi documentation for using Python. This has basic commands and information on Thonny (the recommended IDE – integrated development environment).
magpi.cc/pidocs

DEVDOCS DOCUMENTATION
For most use cases, the official Python documentation and Raspberry Pi documentation will cover everything you need. In some instances, you may find the DevDocs pages cover optional components that aren't part of the official distribution (but are commonly used).
devdocs.io

PYTHON DOCUMENTATION
Make sure you go straight to the horse's mouth and bookmark the Python documentation website.
docs.python.org

Codecademy

There are a dazzling array of interactive coding websites and courses out there, and the truth is that many of them are excellent, free, and will do the job just fine.

Codecademy is not a wholly free option, and many of its more advanced courses are part of its Pro package. There's a lot to learn from its free offerings, though, and we like its interactive web-based interface.

Once you've exhausted the free material, it is worth taking a look at its Pro offering as there are courses in data science, chatbots, and financial analysis. You will also find technical interview practice on the website, which is handy if you're thinking of coding for a living. ◾

Learn Python

If you want to learn Python but aren't too keen on paying for an interactive course, then Learn Python is a good place to start.

The course is supported by DataCamp, which creates interactive Python tutorials for data science. However, this introductory course is wholly free.

Each page covers a common Python topic and features an interactive web-based shell, so you can code along with the examples.

It takes you all the way from the basics, up to more advanced subjects and beginner data analysis. Sadly, there aren't any Raspberry Pi-specific tutorials in the course, but it will serve as a great accompaniment to some of the projects provided by Raspberry Pi itself (**magpi.cc/projects**). ◾
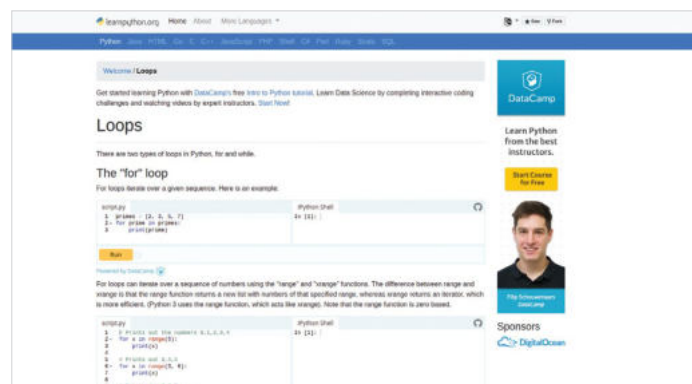
Python books

Prefer to learn with paper? Then pick up these books
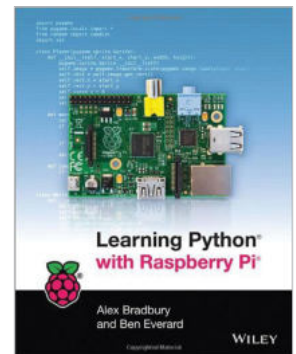
LEARNING PYTHON WITH RASPBERRY PI

Ben Everard, from *The MagPi*'s sister magazine HackSpace, helped write this fantastic book. It takes you from a standing start, and is especially aimed at Raspberry Pi users.
magpi.cc/learningpython

PROGRAMMING THE RASPBERRY PI, SECOND EDITION: GETTING STARTED WITH PYTHON

This book by Simon Monk is another excellent guide that blends Python coding with electronics projects.
magpi.cc/programmingpi

HEAD FIRST PYTHON

Paul Barry's *Head First Python* book is well respected. It lacks specific Raspberry Pi coverage, but its highly visual style makes learning easier on the eyes.
magpi.cc/headfirstpython

THE *Official*

RASPBERRY PI
PROJECTS BOOK
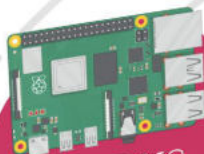
VOLUME 5

INCLUDING
RASPBERRY
PI 4!
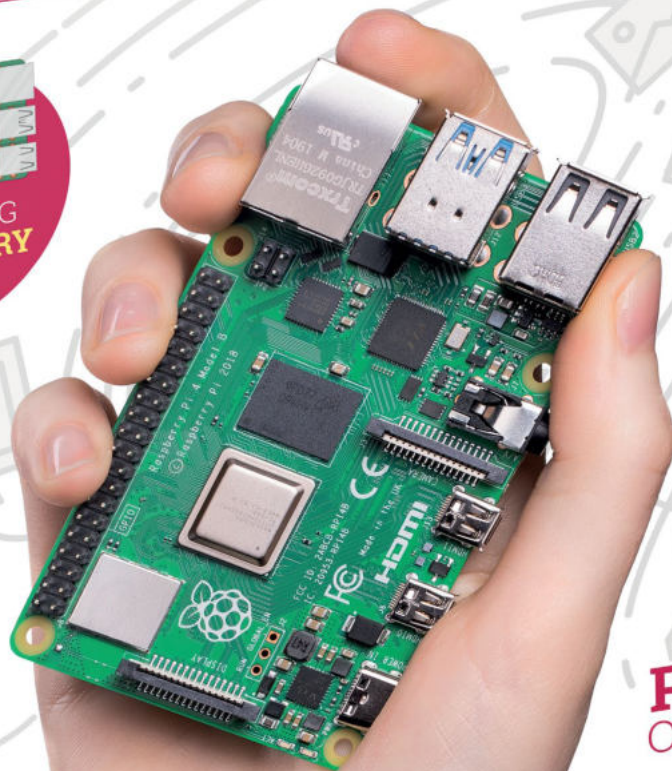
200
PAGES
OF IDEAS

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

THE *Official*

RASPBERRY PI
PROJECTS BOOK

VOLUME 5

200 PAGES OF RASPBERRY PI

- Use Raspberry Pi 4 as a desktop PC and 4K home media centre

- Be inspired by incredible projects made by other people

- Learn how to code and make with our step-by-step tutorials

- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store

Caroline Dunn

A prolific maker and YouTuber focusing on technology tutorials

> Name **Caroline Dunn** | > Community role **YouTuber**
> Occupation **Marketing consultant** | > Website **magpi.cc/cdunn**

▼ Caroline originally made her own AIY kit due to difficulty in getting *The MagPi* #57

"I started my channel by accident, when I posted a video of myself making a Christmas ornament from three wine corks," recalls Caroline Dunn. She makes weekly tech videos for her YouTube channel, several of which include Raspberry Pi. "[It was] called 'How to Make a Reindeer Cork Ornament'. I enjoyed explaining things visually (with my hands), but I was extremely camera shy; I rarely show my face in my videos 2010–2014. In 2016, I noticed how few technology channels were led by females, and I decided to transition my channel to technology to show men, women, and little girls out there that technology is not gender-specific."

Caroline has a background in engineering – she studied it at university and has a master's degree in Electrical and Computer Engineering. However, these days she's a marketing consultant.

What was your first Raspberry Pi project?

I learned about the Raspberry Pi April 2014 during my first project with [it], installing Alexa Voice Service with my friend Steve Youngblood. We saw a tutorial on 'How to Make an Alexa' and decided to try it with no prior Raspberry Pi experience. It took us six hours to complete our first Raspberry Pi project. We would work on [it] every day on our lunch hour at work and invite our co-workers in our office to watch and/or help. Mostly they just laughed at us bumbling our way through the project. Morale at work was low and we provided comic relief. Prior to this project, I had not used terminal commands in well over a decade. We documented the entire experience here: **magpi.cc/diyecho**.

What's been your favourite (Raspberry Pi-related) video to make?

Wow, that's a tough question, like picking a favourite child or pet. I like practical projects, ones where there is a usefulness to our everyday lives. The most useful
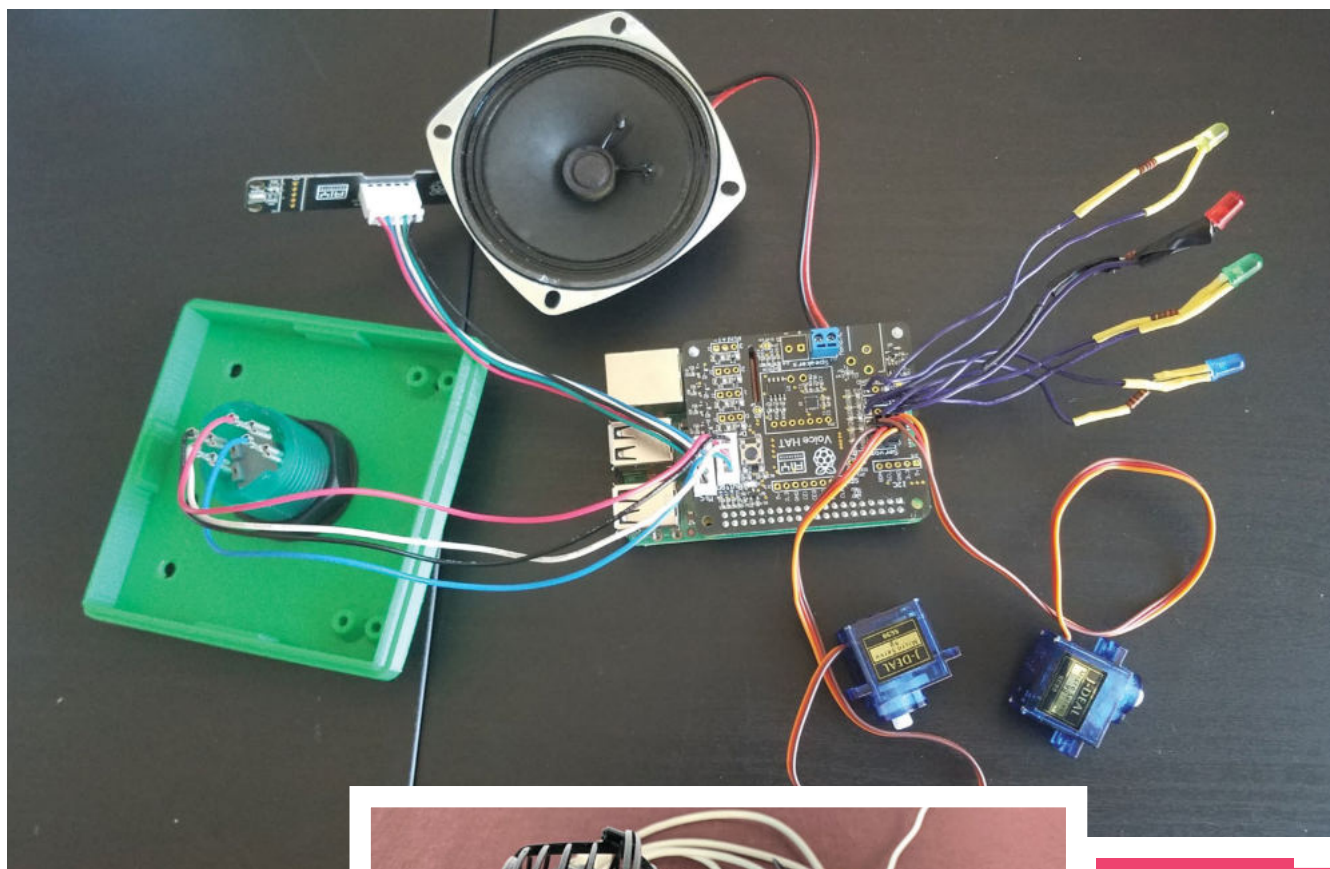
▲ When the AIY Voice Kit went on sale, Caroline finally got to try it out

project that I have made and still use today is the motionEye project with motion-triggered photo and video. I initially set it up to monitor my geriatric cat, but after a large fire in my neighbour's apartment, I use it to monitor anyone coming in and out of my apartment: **magpi.cc/petmonitor**.

> ❝ I like practical projects, ones where there is a usefulness to our everyday lives ❞

What is your story with *The MagPi* issue 57?
When [that issue] came out with the Google AIY kit, I went on a wild goose chase to try to buy [it].

▲ An inventive way to help keep your Raspberry Pi cool

I called every bookstore in my city in an attempt to find the elusive issue 57. One bookstore employee said, "You're only the sixth person to call with that question today," before hanging up on me. I was desperate to do the Google AIY project and started researching ways to complete the project without issue 57. I ordered a 'button' so that I could do the project, then I became impatient waiting two days for it to arrive. While waiting, I figured out how to do the project without the button and without issue 57. When the button arrived, I completed the project with the button. Approximately a year later, the AIY kits were sold at my local Micro Center, and then I finally completed the original project from issue 57: **magpi.cc/diyhome**. M

Mental well-being

"Because I am concerned for the mental well-being of people sheltering at home, I've started a weekly live show on my YouTube channel where anyone can ask me questions, and I present project ideas that can be completed at home. I dedicate a section of my show to Raspberry Pi every week. For example, last week I provided the free link to the *Code the Classics* [book] made possible by Raspberry Pi. Anyone can access my weekly show from my channel page, **youtube.com/ caroline**. My weekly show is broadcast every Thursday, 3pm EST / 8pm GMT."

This Month in
Raspberry Pi

MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: **have you made something with a Raspberry Pi over the weekend?** Every Monday, our followers send us amazing photos and videos of the things they've made. Here is a small fraction of them. Follow along at the hashtag #MagPiMonday. M

01. We love some good upcycling with a vintage radio

02. Very important work for sure

03. Balancing bots are very tricky, but adding legs with full motion seems like a mighty task

04. A lot of people were enamoured by this rack from Peter Mount

05. This is a great use of Raspberry Pi Zero, and an equally great 3D-printed case as well

06. Art but make it 'better' with AI

07. A very pleasing LED board – look out for more in a future issue!

08. Nature photography always looks great

09. We like the ingenuity of this fully featured mobile sensor – it's a bit like a *Star Trek* tricorder

10. Many folks have been experimenting with the new High Quality Camera

Michael (Mike) Horne
@recantha

Replying to @TheMagPi

Went to robot club with @CannonFodder :-) Also started the teardown of a vintage ish radio to convert it for Pi.

10:52 AM · May 11, 20

Dr Footleg - Roboteer
@drfootleg

Replying to @TheMagPi

I spent the best part of 2 days modelling a cats bum.

Dr Footleg - Roboteer @drfootleg · May 17
Replying to @drfootleg @alister_ware and @piofthings
Hard to believe it has taken me over a day to remodel this cat's bum. Specifically this tail servo mounting bracket. But I think you'll agree it looks very fine.

10:28 AM · May 18, 2020 · Twitter Web App

04

Peter Mount
@peter_mount

Replying to @TheMagPi

Made more progress on the 3d printed racking.

I've got the 1.14" tft screen working under raspbian & have Ubuntu 64bit installed. Got to install the screen to that pi later this week as well as cable & power management
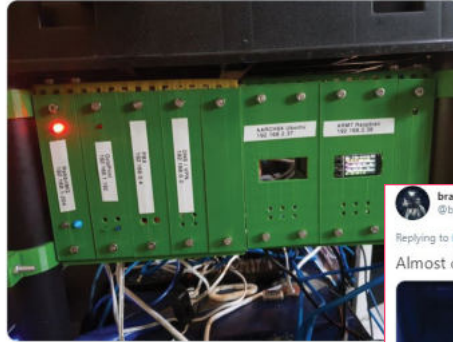
10:38 AM · May 4, 2020 from Maidstone, South East · Twitter for Android

03

Raspibotics
@raspibotics

Replying to @TheMagPi

Progress on the balancing robot. Got inverse kinematics working on the robot legs so that I can adjust the height of the robot when I get the balancing working.

2:35 PM · May 4, 2020 · Twitter for Android

07

braghettos
@braghettos

Replying to @TheMagPi

Almost done!!

· May 14, 2020 · Twitter for Android

05

TGD
@TGD_Consulting

Replying to @TheMagPi

Finished #3Dprinted housing for CO2, particulate matter monitor based on PiZeroWH & @Pimoroni inky display.

All 3D-models are available @Cults3D
cults3d.com/en/3d-model/ga...

#IoT #RaspberryPi #monitoring #environment
#CitizenScience #AirQuality

11:17 AM ·

06

8 Bits and a Byte
@8BitsandaByte

Replying to @TheMagPi

A pi powered, internet controllable robot, that generates fine art from user comments.

You can try it yourself:
remo.tv/join/rffrwxr

May 11, 2020 · Twitter Web App

09

Ciprian Manea
@ovipic

Replying to @TheMagPi

We @unpi_ong made a "mobile measuring stick" to test outside our sonar and laser distance sensors for the upcoming #PiWars #unPi @PiWarsRobotics @pimoroni
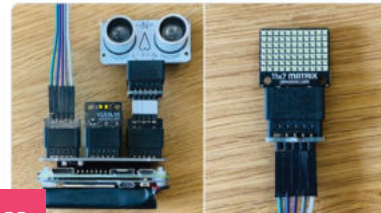
M · May 18, 2020 · Tweetbot for iOS

08

Leo White
@LeoWhitesTweets

Replying to @TheMagPi

Finished settings up my @naturebytesuk case (Which has been on the shelf for a few years) using the PI-TIMOLO software.

So far spotted a fox, cat, various birds, moths and a red, glowey... thing?

May 18, 2020 · Twitter Web App

10

David Booth
@Worldv6440

Replying to @TheMagPi

Hi, I've been putting the new @Raspberry_Pi HQ camera through its paces by making a PiDSLR. I've done a little write-up hosted on @recantha 's blog:

recantha.co.uk/blog/?p=20194

11:03 AM · May 18, 2020 · Twitter Web App

Shot on Raspberry Pi

More great photos taken with the High Quality Camera

We put a nice selection of shots taken with the HQ Camera in our big feature on page 66, but more kept rolling in so we decided to show them off here. M
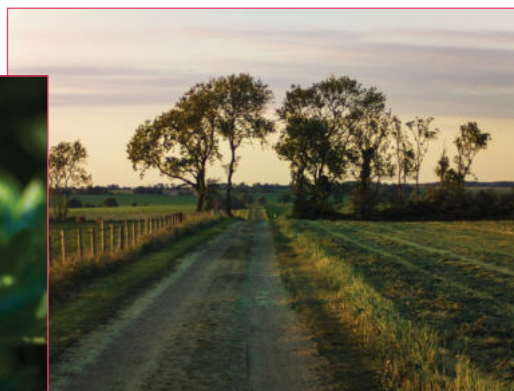
▲ **David Booth** (@Worlds6440)

"I've been putting the new Raspberry Pi HQ camera through its paces by making a PiDSLR."

▶ **JAKUBZ** (@terramyx)

"Farthest and closest objects I shot using RPi so far. Tair-3 300/4.5 lens for astro, 60 minutes total exposure time. CZJ Flektogon 35/2.4 + extension tube for macro. I'm impressed by this little camera and can't wait to find new usages for it."
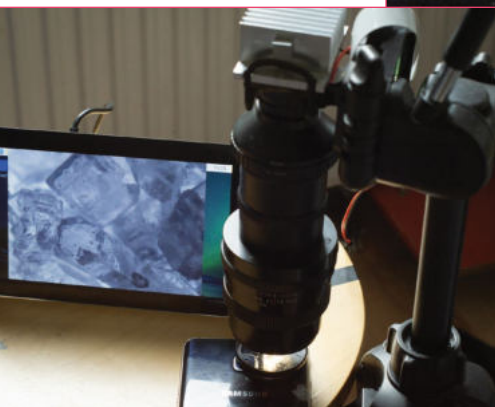
▲ **Colin Alston** (@colinza)

"My 3D-printed lens mount works perfectly! So excited by this."

Crowdfund **this!**

Raspberry Pi projects you can crowdfund this month
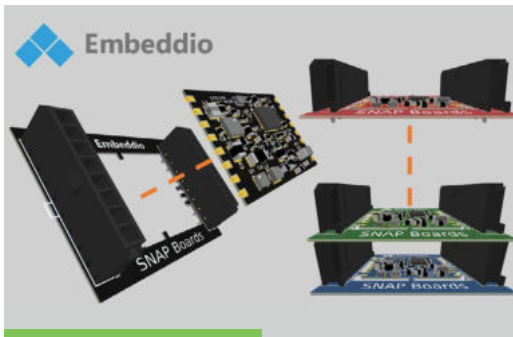
SNAP Boards

These stacking boards allow you to develop circuits and components with a multitude of microcomputers, like Raspberry Pi – and microcontrollers, like Arduino – without having to completely rewire it.

▶ **kck.st/3g1Omkt**

K100 Multitop Keyboard Computer

This project is a cool custom case for a Raspberry Pi that turns it into a keyboard – or at least attaches a keyboard to it. You can then add power, a monitor, and a mouse and get a full traditional desktop setup.

▶ **kck.st/2ylPo9P**

CROWDFUNDING A PROJECT?
If you've launched a Raspberry Pi-related project, let us know!
magpi@raspberrypi.org

Raspberry Jam
Event Calendar

Find out what community-organised Raspberry Pi-themed events are happening near you...

01. Moon Hack 2020
📅 Saturday 30 May
📍 Online
▶ **magpi.cc/TNZg8m**

A world-record-breaking online event for kids who love to code.

02. Preston Raspberry Jam
📅 Monday 1/15 June
📍 Online
▶ **magpi.cc/TUBrfq**

An evening of online talks and networking for people with an interest in Raspberry Pi.

03. Free Robotics Sunday Virtual Lesson
📅 Thursday 4 June
📍 Online
▶ **magpi.cc/6K7br1**

Free after-school robotics near Roppongi, Tokyo, for teens, that takes place every three days.

04. Introduction to Electronics and Programming with the Raspberry Pi
📅 Saturday 6 June
📍 Online
▶ **magpi.cc/mqhYHn**

Learn the fundamentals of electronics and programming and build your own project in this course.

05. West Sound CoderDojo on Zoom!
📅 Saturday 6/20 June
📍 Online
▶ **magpi.cc/q2EyI9**

Ninjas, parents, mentors, and all are invited to join West Sound CoderDojo for bi-weekly sessions of coding and fun.

06. Houston Raspberry Pi Meetup
📅 Wednesday 10 June
📍 Online
▶ **magpi.cc/qfSx5y**

A community of coders, artists, educators, and engineers interested in project-based learning with the Raspberry Pi platform.

07. Medway Makers Tech Meetup
📅 Sunday 14 June
📍 Online
▶ **magpi.cc/eTFi0C**

Workshops in building cool things with Arduino, how to code for beginners, how to use a Raspberry Pi, and so on.

08. CoderDojo Oostende
📅 Saturday 20 June
📍 Online
▶ **magpi.cc/I5JbxW**

This Belgium CoderDojo will be running online for people who speak Dutch.

FULL CALENDAR

Get a full list of upcoming events for June and beyond here:

rpf.io/jam

FIND OUT ABOUT JAMS

Want a Raspberry Jam in your area? Want to start one?

jam@raspberrypi.org

We've highlighted some of the areas in need of a Jam! Can you help out?

Raspberry Jam advice:
Mailing lists

"**H**aving a staff mailing list for all your active volunteers really helps: mailmanlists.net run email discussion lists, and they give free accounts to worthy causes like Raspberry Jams."

Andrew Oakley – Cotswold Jam

Every Raspberry Jam is entitled to apply for a Jam starter kit, which includes magazine issues, printed worksheets, stickers, flyers, and more. Get the book here: **rpf.io/guidebook**

RASPBERRY JAM GUIDEBOOK
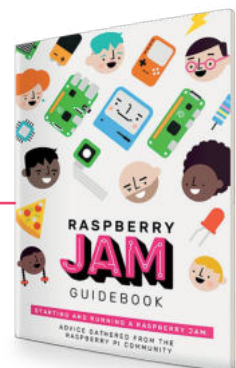
Your Letters
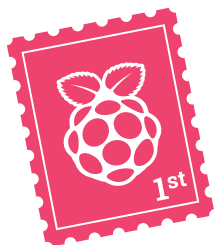
New camera projects

I got the new High Quality camera for my Raspberry Pi and have been snapping away with photos. Are you interested to see the photos I've been taking, or even the camera itself that I made?

Kylie via email

We'd love to see photos anyone has taken with the camera, and we'd also love to see any projects you've created that use the camera. It's always a good idea to put these photos on Twitter using the #ShotOnRaspberryPi hashtag, as that way the folks at Raspberry Pi can see them as well.

You may also want to check our big feature on projects for the High Quality camera, which you can find starting on page 66.

Submit photos and projects to us at **magpi@raspberrypi.com** and we'll try to get them in the magazine.

▶ Fancy turning a Raspberry Pi into a point-and-shoot camera? We have a tutorial starting on page 70
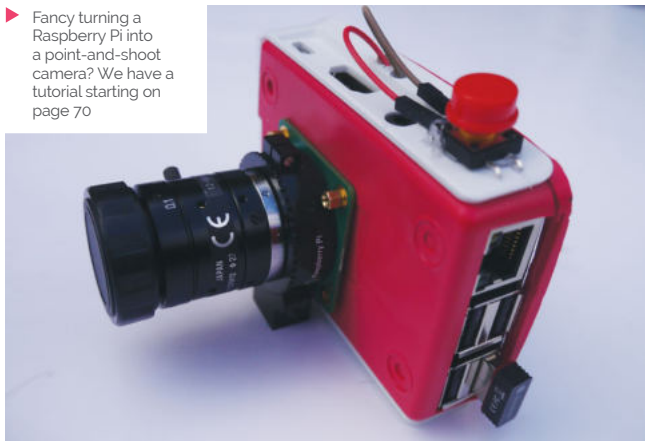
Mount question

I've been thinking about getting one of those new HQ Cameras for my Raspberry Pi. I read about it having interchangeable lenses, but they look pretty small – would lenses from a Nikon camera work with it? Would be nice to have a variety.

Charles via Facebook

The High Quality Camera has a CS lens mount and is supplied with a C–CS adapter, so it will work with any C- or CS-mount lens. Unfortunately, Nikon lenses use an F mount – you can get adapters to use them with a C/CS mount, although they're not cheap. Camera lens kit rarely is, though.

We've seen people use 3D-printed adapters as well, but we can't speak for their quality.

▶ If you want to learn more about the new High Quality camera, check out our camera book at **magpi.cc/cameraguide**
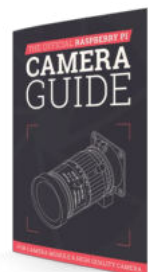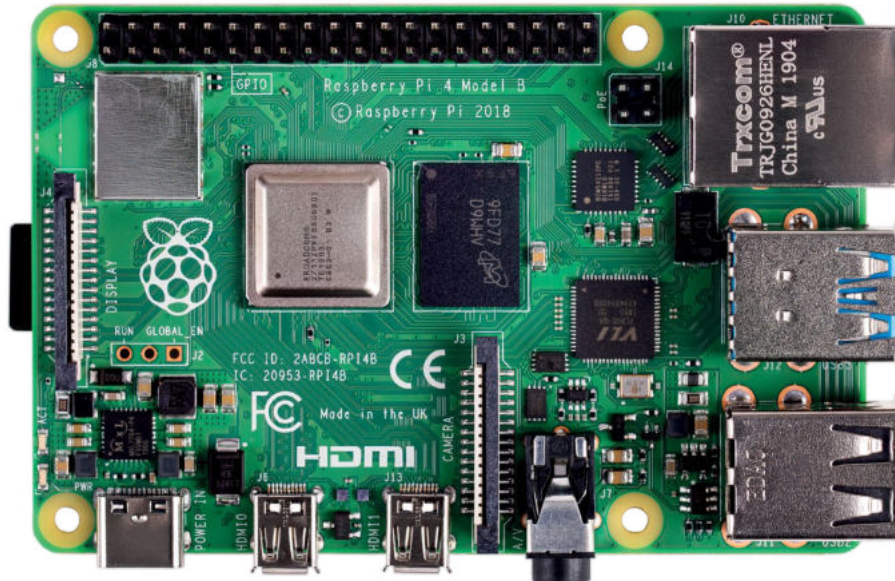
▶ While Raspberry Pi 4 does have a USB-C port, it requires far more power

Coolest Projects where?

I was wondering where and when Coolest Projects would be held this year? My daughter was excited to go to the International one this year.

Peter via email

This year, Coolest Projects is going online instead of having separate events in Manchester and Dublin. Anyone up to the age of 18 can register to submit a project entirely from home for the competition, and you can find out loads more details at **magpi.cc/cponline**.

▲ Take your skills online for Coolest Projects 2020

The most portable Raspberry Pi

Have you ever seen (or even written about?) projects where users connect a Raspberry Pi to their phone so they have Linux on the go?

As far as I know, Raspberry Pi can be powered over a direct connection from a USB-C port for example. So with three pieces, one would have mobile Linux: a phone with USB-C, a USB-C-to-micro-USB adapter, and a Raspberry Pi. Then one could SSH in to Raspberry Pi from the phone.

Marek via Twitter

We've not seen anyone attempt a project quite like this, but we've seen similar stuff that uses mobile batteries and phone hotspots, or even plugging a Raspberry Pi Zero into a laptop in gadget mode.

The main issue we can see with this is that most phones don't really allow you to power devices via their USB-C port, and definitely not up to the 2 A of current you'd need for a Raspberry Pi. Even using a very minimal Raspberry Pi Zero installation would be too much.

It's an interesting idea but would probably require a little more hacking of your phone to do some funky things with its USB-C port, which we can't imagine would be good for the phone. Might as well stick to the portable battery.

Contact us!

> Twitter   **@TheMagPi**
> Facebook   **magpi.cc/facebook**
> Email   **magpi@raspberrypi.com**
> Online   **raspberrypi.org/forums**

WIN ONE OF TEN
MAKER PHATS!

Simplify learning electronics with the Maker pHAT, allowing for quick circuit building and programming. We have ten to give away...
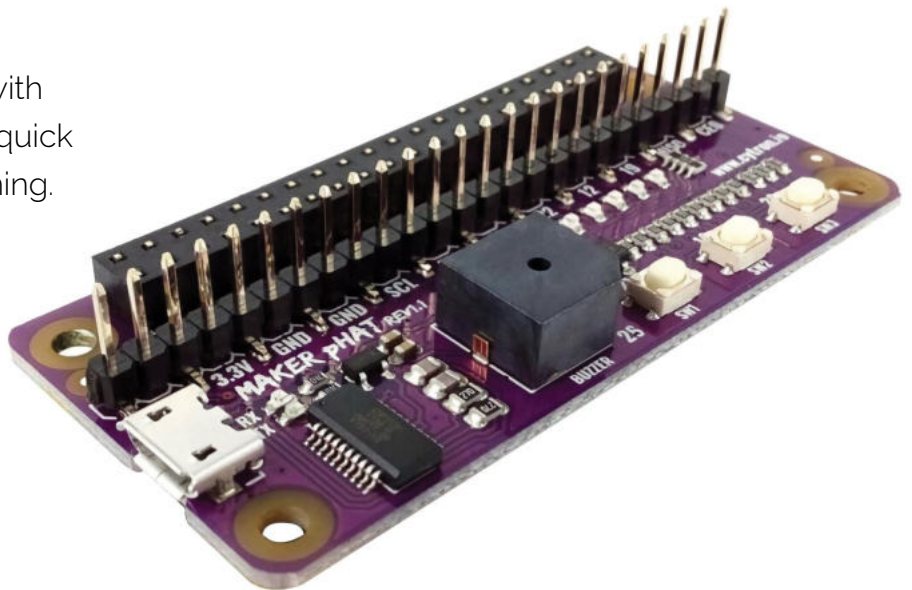
Head here to enter: magpi.cc/win | **Learn more: magpi.cc/makerphat**

Terms & Conditions

Competition opens on **27 May 2020** and closes on **25 June 2020**. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram or Facebook.

RetroPie

WITH RASPBERRY PI 4

EASY EMULATION WITH THE ULTRA-POWERFUL RASPBERRY PI 4

THE MAGPI #95 ON SALE 25 JUNE

Plus!

Learn computing with Raspberry Pi

Monitor your energy usage

Build an ISS tracker with Inky wHAT

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER	**@TheMagPi**
FACEBOOK	**fb.com/MagPiMagazine**
EMAIL	**magpi@raspberrypi.com**

Get ready for the recovery

Raspberry Pi is stepping up to the challenge and making a difference. By **Eben Upton**

When I agreed to write this column, back in the first week of March, I was pretty sure I knew what it would be about. The team had spent a month struggling with the aftermath of the 'long Chinese New Year': factories in China were at less than half capacity, leading to delays in the introduction of the High Quality Camera (due to a shortage of lenses) and the 8GB Raspberry Pi 4 (due to delays

❝ We're doing everything we can to make a difference ❞

qualifying an inductor for the new Dialog switch-mode power supply you can see in the bottom left of the board).

I would have reflected on the inevitable vulnerability of businesses like ours, with globe-spanning supply chains, to unexpected events in far-away countries. And I might have been tempted to come across a little smug about our UK manufacturing base, vendor diversity, and comfortable inventory position. What a difference two months makes.

Working to help

With almost every advanced industrial economy in some variation of lockdown, the only story worth telling is what Raspberry Pi has been doing, in some small way, to help mitigate the impact on people's lives.

Most directly, we've been working to support companies around the world who are designing ventilators powered by Raspberry Pi. Sadly, none of the teams we've worked with were selected for the UK's Rapidly Manufactured Ventilator Systems program, but globally we've shipped over 20,000 Raspberry

Pi Zero computers into ventilator applications. And we've been using our component sourcing operation to bring in much-needed PPE for parts of the UK health service.

In the UK, over 800,000 schoolchildren are being asked to study remotely, despite not having a computer in their household. We're working with philanthropists and implementation partners to get Raspberry Pi 4 computers into the hands of some of these young people. Our goal in starting Raspberry Pi was to provide children with a platform to learn about computers; for these children our computers have become their only route to any kind of education.

The Foundation has been working hard to create content to support teachers and learners in this new world, and to find ways that organisers of Code Clubs and CoderDojos can continue to run their clubs online. Every online course we run is seeing record enrolments and, crucially, people are completing their courses in record numbers, and in record time.

And even though the shops are shut, our publications – from Hello World and *The MagPi*, to HackSpace and Wireframe – are still available online for those who want to learn something new at home. Across the organisation, we've made the choice to step up to the challenge, not to hunker down and wait for it to pass: this isn't the crisis we were set up to solve, but it's the crisis we've been given, and we're doing everything we can to make a difference. I'm incredibly proud of the team, and of you, our community. We're going to come through this together, and we're going to come through it stronger.

Get ready for the recovery: 2021 is going to be a hell of a year. **M**

AUTHOR

Eben Upton

Eben is the creator of Raspberry Pi and a founder of the Raspberry Pi Foundation.

magpi.cc/eben
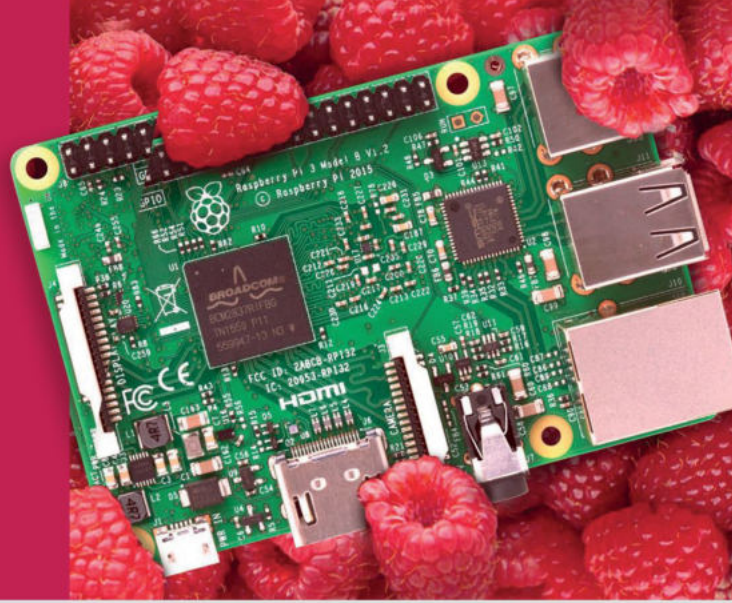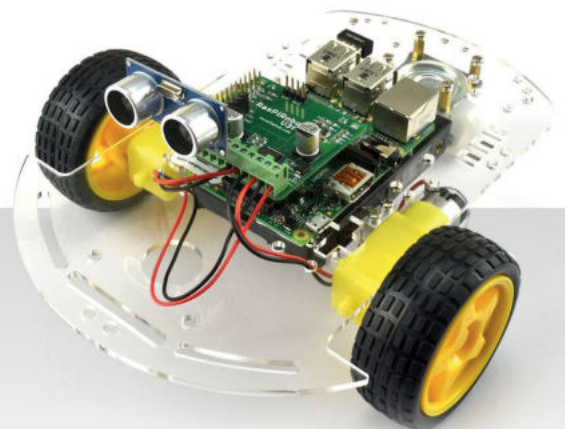
RASPBERRY PI 4 STARTER KIT

This 8-piece kit includes everything you need to get going in the world of Raspberry Pi– from all the hardware in a single box, to access to an online step-by-step guide as well as projects to get up and running faster than ever

okdo™
DESIGN THE WORLD